**UNIVERSITAS GUNADARMA**

**FAKULTAS ILMU KOMPUTER & TEKNOLOGI INFORMASI**

****

**TULISAN ILMIAH**

**PEMETAAN KEPADATAN PENDUDUK DKI JAKARTA MENGGUNAKAN K-MEANS CLUSTERING UNTUK OPTIMALISASI PROGRAM PEMERINTAH**

**Nama : Annisa Rahmaningsih**

**NPM : 10121181**

**Jurusan : Sistem Informasi**

**Pembimbing : Dr. Lulu Chaerani Munggaran, SKom., MMSI.**

**Diajukan Guna Melengkapi Sebagian Syarat Dalam Mencapai Gelar Setara Sarjana Muda**

**JAKARTA**

**2024**

# **PERNYATAAN ORISINALITAS DAN PUBLIKASI**

Saya yang bertanda tangan di bawah ini,

Nama : Annisa Rahmaningsih

NPM : 10121181

Judul Tulisan Ilmiah : PEMETAAN KEPADATAN PENDUDUK DKI JAKARTA MENGGUNAKAN K-MEANS CLUSTERING UNTUK OPTIMALISASI PROGRAM PEMERINTAH

Tanggal Sidang : 02 September 2024

Tanggal Lulus :

menyatakan bahwa tulisan ini adalah merupakan hasil karya saya sendiri dan dapat dipublikasikan sepenuhnya oleh Universitas Gunadarma. Segala kutipan dalam bentuk apa pun telah mengikuti kaidah, etika yang berlaku. Mengenai isi dan tulisan adalah merupakan tanggung jawab Penulis, bukan Universitas Gunadarma. Demikian pernyataan ini dibuat dengan sebenarnya dan dengan penuh kesadaran.

****Jakarta, 22 Agustus 2024

(Annisa Rahmaningsih)

# **LEMBAR PENGESAHAN**

Judul PI : Pemetaan Kepadatan Penduduk DKI Jakarta Menggunakan K-Means Clustering Untuk Optimalisasi Program Pemerintah

Nama : Annisa Rahmaningsih

NPM : 10121181

Tanggal Sidang : 02 September 2024

Tanggal Lulus :

Menyetujui

Pembimbing Kasubag. Sidang PI

(Dr. Lulu Chaerani.M, SKom., MMSI) (Dr. Sri Nawangsari, SE., MM. M.I.kom.)

Ketua Jurusan

(Dr. Setia Wirawan, SKom, MMSI.)

# **ABSTRAK**

Annisa Rahmaningsih, 10121181

PEMETAAN KEPADATAN PENDUDUK DKI JAKARTA MENGGUNAKAN K-MEANS CLUSTERING UNTUK OPTIMALISASI PROGRAM PEMERINTAH

Tulisan Ilmiah. Sistem Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi. Universitas Gunadarma. 2025.

Kata Kunci : Data Mining, K-Means, Clustering, Pemetaan

(xv + 62 + Lampiran)

DKI Jakarta, sebagai ibu kota negara Indonesia, memainkan peran yang penting dalam berbagai aspek kehidupan nasional, mulai dari ekonomi, politik, hingga sosial budaya. Sebagai pusat pemerintahan dan bisnis, Jakarta menarik banyak penduduk dari berbagai daerah di Indonesia, yang datang untuk mencari peluang kerja dan kehidupan yang lebih baik. Tujuan dari penelitian ini adalah untuk mengimplementasikan *clustering* menggunakan algoritma K-Means untuk pemetaan kepadatan penduduk DKI Jakarta agar membantu pemerintah mengoptimalisasi program-program yang ada. Metode penelitian *clustering* dengan algoritma K-Means digunakan untuk membagi data menjadi beberapa kelompok. Jumlah *cluster* total ada 3 yaitu *cluster* 1 untuk jumlah wilayah dengan penduduk rendah, *cluster* 2 dengan jumlah penduduk sedang, dan *cluster* 3 untuk jumlah penduduk padat kemudian dilakukan analisis terhadap pengelompokkan tersebut. Data kepadatan penduduk DKI Jakarta yang digunakan diambil dari situs resmi <https://satudata.jakarta.go.id/home>, dan tahapan pembuatan program dilakukan dengan menggunakan Google Colab. Hasil analisis menunjukan bahwa setiap cluster memiliki karakteristik dan kebutuhan yang berbeda, cluster 1 membutuhkan peningkatan infrastruktur dasar, dan aksebilitas. Cluster 2 membutuhkan pengembangan infrastruktur sekunder dan pengelolaan tata ruang, cluster 3 memerlukan pengelolaan kepadatan penduduk, peningkatan infrastruktur, serta kontrol lingkungan. Kesimpulanya, pemetaan ini memberikan informasi yang berguna bagi pemerintah untuk mengoptimalkan kebijakan dan program pembangunan sesuai dengan kebutuhan masing-masing cluster.

Daftar Pustaka (2017-2023)

# **ABSTRACT**

Annisa Rahmaningsih, 10121181

POPULATION DENSITY MAPPING OF DKI JAKARTA USING K-MEANS CLUSTERING FOR GOVERNMENT PROGRAM OPTIMIZATION

Scientific Paper. Information Systems, Faculty of Computer Science and Information Technology, Gunadarma University. 2025.

Keyword : Data Mining, K-Means, Clustering, Mapping

(xv + 62 + Appendix)

DKI Jakarta, as the capital city of Indonesia, plays an important role in various aspects of national life, including the economy, politics, and socio-culture. As the center of government and business, Jakarta attracts many people from different regions of Indonesia who come in search of job opportunities and a better life. The aim of this research is to implement clustering using the K-Means algorithm to map the population density of DKI Jakarta, helping the government optimize existing programs. The K-Means clustering method was used to divide the data into several groups. The total number of clusters is 3: cluster 1 for areas with low population, cluster 2 with medium population, and cluster 3 with high population, followed by analysis of the grouping. The population density data for DKI Jakarta was obtained from the official site <https://satudata.jakarta.go.id/home>, and the program was developed using Google Colab. The analysis results show that each cluster has different characteristics and needs: cluster 1 requires improvements in basic infrastructure and accessibility, cluster 2 needs secondary infrastructure development and urban planning, while cluster 3 requires population density management, infrastructure improvements, and environmental control. In conclusion, this mapping provides useful information for the government to optimize policies and development programs according to the needs of each cluster.

Bibliography (2017-2023)

# **KATA PENGANTAR**

Segala puji dan Syukur ke hadirat Tuhan Yang Maha Kuasa yang telah memberikan berkat, anugerah dan karunia yang melimpah, sehingga penulis dapat menyelesaikan Penulisan Ilmiah ini. Penulisan Ilmiah ini disusun guna melengkapi sebagian syarat dalam mencapai gelar Setara Sarjana Muda pada Jurusan Sistem Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Gunadarma. Adapun judul Penulisan Ilmiah ini adalah Pemetaan Kepadatan Penduduk DKI Jakarta Menggunakan K-Means Clustering Untuk Optimalisasi K-Means Clustering.

Walaupun banyak kesulitan yang penulis harus hadapi ketika Menyusun Penulisan Ilmiah ini, namun berkat bantuan dan dorongan dari berbagai pihak akhirnya tugas ini dapat diselesaikan dengan baik. Untuk itu penulis mengucapkan terima kasih, kepada :

1. Prof. Dr. E.S. Margianti, SE., MM., selaku Rektor Universitas Gunadarma.
2. Prof. Dr.rer-nat Achmad Benny Mutiara, selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Gunadarma.
3. Dr. Sri Nawangsari, SE., MM., selaku Kepala Sub Bagian Sidang Penulisan Ilmiah Universitas Gunadarma.
4. Ibu Dr. Lulu Chaerani Munggaran, SKom., MMSI. selaku Dosen Pembimbing yang senantiasa membimbing, memberikan arahan, saran, dan waktunya kepada penulis selama proses pembuatan Penulisan Ilmiah ini dari awal hingga akhir
5. Bapak Asiyanto dan Ibu Chairiana, selaku orang tua penulis yang telah memberikan motivasi dan dukungan secara moril dan materil sehingga penulis dapat menyelesaikan Penulisan Ilmiah ini.
6. Maisarah Salwa Arief Saputri, selaku sahabat penulis yang sudah memberikan dukungan materil untuk penulisan ini
7. Semua pihak yang telah memberikan dukungan dan dorongan baik secara langsung maupun tidak langsung yang penulis tidak dapat sebutkan namanya satu persatu.
8. Untuk diri sendiri, terimakasih sudah mampu menyelesaikan penulisan ini sampai akhir.

Semoga Allah SWT. membalas budi dan jasa semua pihak yan telah membantu dalam hal menyelesaikan penulisan ilmiah ini. Akhir kata, semoga penulisan ini dapat bermanfaat bagi semua pihak, termasuk penulis pada khususnya dan pembaca pada umumnya.

Jakarta, 22 Agustus 2024

( Annisa Rahmaningsih )

# **DAFTAR ISI**

[**PERNYATAAN ORISINALITAS DAN PUBLIKASI** i](#_Toc177459979)

[**LEMBAR PENGESAHAN** ii](#_Toc177459980)

[**ABSTRAK** iii](#_Toc177459981)

[**ABSTRACT** iv](#_Toc177459982)

[**KATA PENGANTAR** v](#_Toc177459983)

[**DAFTAR ISI** vii](#_Toc177459984)

[**DAFTAR GAMBAR** ix](#_Toc177459985)

[**DAFTAR TABEL** xi](#_Toc177459986)

[**1.** **PENDAHULUAN** 1](#_Toc177459987)

[**1.1 Latar Belakang** 1](#_Toc177459988)

[**1.2 Ruang Lingkup** 2](#_Toc177459989)

[**1.3 Tujuan Penelitian** 2](#_Toc177459990)

[**1.4 Metode Penelitian** 2](#_Toc177459991)

[**1.5 Sistematika Penulisan Ilmiah** 4](#_Toc177459992)

[**2.** **TINJAUAN PUSTAKA** 5](#_Toc177459993)

[**2.1 Data Mining** 5](#_Toc177459994)

[**2.2 Pengenalan Pola Data Mining dan Machine Learning** 6](#_Toc177459995)

[**2.2.1Tahap-tahap data mining** 7](#_Toc177459996)

[**2.2.2Metode Data Mining**  9](#_Toc177459997)

[**2.3 Pengklasteran (*Clustering*)** 12](#_Toc177459998)

[**2.4 Analisis Karakteristik Metode Clustering** 13](#_Toc177459999)

[**2.5 Algoritma K-Means** 15](#_Toc177460000)

[**2.5.1 Perbandingan K-Means dengan algoritma clustering lainnya** 18](#_Toc177460001)

[**2.5.2 Kelebihan algoritma K-Means** 19](#_Toc177460002)

[**2.6 Flowchart K-Means** 20](#_Toc177460003)

[**2.7 Bahasa Pemrograman Python** 21](#_Toc177460004)

[**2.7.1 Fitur Python** 22](#_Toc177460005)

[**2.7.2 Library Python** 22](#_Toc177460006)

[**2.8 Pengujian Sillhoutte Score** 25](#_Toc177460007)

[**2.9 Google Colaboratory** 26](#_Toc177460008)

[**2.10 Kepadatan Penduduk** 30](#_Toc177460009)

[**2.11 Satu Data Jakarta** 31](#_Toc177460010)

[**2.12 DKI Jakarta** 5](#_Toc177460011)

[**3.** **PEMBAHASAN** 30](#_Toc177460012)

[**3.1 Gambaran Umum Penerapan *Clustering* Data Mining**  30](#_Toc177460013)

[**3.2 Data Selection** 30](#_Toc177460014)

[**3.3 Pre-processing/cleaning** 32](#_Toc177460015)

[**3.3.1 Import Library** 33](#_Toc177460016)

[**3.3.2 Data mentah** 33](#_Toc177460017)

[**3.3.3 Visualisasi data** 35](#_Toc177460018)

[**3.4 Data Transformation** 37](#_Toc177460019)

[**3.4.1 Inisialisasi kolom kecamatan ke dalam data numerik** 37](#_Toc177460020)

[**3.4.2 *Drop* atribut yang tidak dibutuhkan** 38](#_Toc177460021)

[**3.4.3 Transformasi Data Menjadi Bentuk Array** 39](#_Toc177460022)

[**3.5 Tahap Data Mining-Clustering** 40](#_Toc177460023)

[**3.5.1 Penetapan Fungsi Euclidean Distance dan Algoritma K-Means** 40](#_Toc177460024)

[**3.5.2 Proses Iterasi** 42](#_Toc177460025)

[**3.5.2.1 Proses iterasi (pengulangan) ke-1** 42](#_Toc177460026)

[**3.5.2.2 Proses Iterasi (pengulangan) ke-2** 45](#_Toc177460027)

[**3.5.2.3 Proses iterasi ke-3** 47](#_Toc177460028)

[**3.6 Pengujian Sillhoutte Score** 50](#_Toc177460029)

[**3.7 Pembentukan ulang data** 50](#_Toc177460030)

[**3.8 Analisis perhitungan menggunakan algoritma K-Means** 52](#_Toc177460031)

[**4.** **PENUTUP** 54](#_Toc177460032)

[**4.1 Kesimpulan** 54](#_Toc177460033)

[**4.2 Saran** 54](#_Toc177460034)

[**DAFTAR PUSTAKA** 55](#_Toc177460035)

[**LAMPIRAN** 1](#_Toc177460036)

# **DAFTAR GAMBAR**

[Gambar 2. 1 Hubungan ilmu data mining dan visualisasi data 7](#_Toc176084617)

[Gambar 2. 2 Proses tahapan KDD 8](#_Toc176084618)

[Gambar 2. 3 Metode data mining 9](#_Toc176084619)

[Gambar 2. 4 Flowchart K-Means 19](#_Toc176084620)

[Gambar 2. 5 Tampilan awal Google Colab 25](#_Toc176084621)

[Gambar 2. 6 Contoh input kode program 25](#_Toc176084622)

[Gambar 2. 7 Pengaturan Google Colab 26](#_Toc176084623)

[Gambar 2. 8 Input kode untuk koneksi drive 26](#_Toc176084624)

[Gambar 2. 9 Tampilan berhasil mengakses drive 26](#_Toc176084625)

[Gambar 2. 10 Daftar file dan folder 27](#_Toc176084626)

[Gambar 2. 11 Input kode untuk upload file 27](#_Toc176084627)

[Gambar 2. 12 Tampilan menu setalan Google Colab 28](#_Toc176084628)

[Gambar 3. 1 Tahapan penulisan menggunakan KDD 30](#_Toc175774285)

[Gambar 3. 2 Data penduduk provinsi DKI Jakarta dari satudata.jakarta.go.id 31](#_Toc175774286)

[Gambar 3. 3 Data yang sudah di download dengan format .CSV 31](#_Toc175774287)

[Gambar 3. 4 Mengunggah data .CSV ke Google Colab 32](#_Toc175774288)

[Gambar 3. 5 Kode Import Library 33](#_Toc175774289)

[Gambar 3. 6 Kode yang digunakan untuk menampilkan beberapa baris data 34](#_Toc175774290)

[Gambar 3. 7 Tampilan data mentah .CSV 34](#_Toc175774291)

[Gambar 3. 8 Grafik visualisasi data 36](#_Toc175774292)

[Gambar 3. 9 Kode untuk mengurutkan data 36](#_Toc175774293)

[Gambar 3. 10 Grafik data setelah diurutkan 37](#_Toc175774294)

[Gambar 3. 11 Kode untuk menghapus kolom yang tidak relevan 39](#_Toc175774295)

[Gambar 3. 12 Hasil drop kolom periode\_data 39](#_Toc175774296)

[Gambar 3. 13 Hasil transformasi data ke array 2D 40](#_Toc175774297)

[Gambar 3. 14 Kode rumus fungsi Euclidean 41](#_Toc175774298)

[Gambar 3. 15 Hasil perhitungan jarak Euclidean 42](#_Toc175774299)

[Gambar 3. 16 Grafik hasil iterasi ke-1 43](#_Toc175774300)

[Gambar 3. 17 Jarak dan pengelompokan objek data ke centroid iterasi ke-1 44](#_Toc175774301)

[Gambar 3. 18 Hasil perhitungan centroid baru iterasi ke-1 45](#_Toc175774302)

[Gambar 3. 19 Grafik hasil iterasi ke-2 46](#_Toc175774303)

[Gambar 3. 20 Hasil perhitungan centroid baru iterasi ke-2 47](#_Toc175774304)

[Gambar 3. 21 Grafik hasil iterasi ke-3 48](#_Toc175774305)

[Gambar 3. 22 Hasil perhitungan centroid baru iterasi ke-3 49](#_Toc175774306)

[Gambar 3. 23 Hasil pengujian sillhouette score 50](#_Toc175774307)

[Gambar 3. 24 Hasil pembentukan ulang data 51](#_Toc175774308)

[Gambar 3. 25 Jumlah masing-masing cluster 51](#_Toc175774309)

# **DAFTAR TABEL**

Tabel 2. 1 Perbandingan antara K-Means, Fuzzy C-Means, dan K-Medoids………….. 18

[Tabel 3. 1 Atribut yang digunakan 32](#_Toc175774540)

[Tabel 3. 2 Hasil Inisialisasi kolom Kecamatan 38](#_Toc175774541)

[Tabel 3. 3 Hasil masing-masing cluster 52](#_Toc175774542)

# **PENDAHULUAN**

## **1.1 Latar Belakang**

DKI Jakarta, sebagai ibu kota negara Indonesia, memainkan peran yang penting dalam berbagai aspek kehidupan nasional, mulai dari ekonomi, politik, hingga sosial budaya. Sebagai pusat pemerintahan dan bisnis, Jakarta menarik banyak penduduk dari berbagai daerah di Indonesia, yang datang untuk mencari peluang kerja dan kehidupan yang lebih baik. Dengan populasi yang terus bertambah dan semakin beragam, kota ini menghadapi berbagai tantangan yang kompleks dalam hal pengelolaan dan pemberdayaan masyarakat.

Salah satu tantangan utama yang dihadapi oleh pemerintah DKI Jakarta adalah bagaimana mengoptimalkan program-program pemerintah agar dapat menjawab kebutuhan dan permasalahan masyarakat dengan lebih efektif dan efisien. Jakarta yang memiliki populasi besar dan beragam memerlukan perencanaan dan implementasi program pemerintah, seperti bantuan sosial, layanan Kesehatan, Pendidikan, dan pelatihan kerja, harus dirancang dan diimplementasikan secara tepat sasaran. Bantuan ini harus menjangkau Masyarakat yang benar-benar membutuhkan, layanan kesehatan harus berkualitas dan mudah di jangkau, pendidikan harus inklusif dengan fasilitas yang memadai, dan pelatihan kerja harus relevan dengan kebutuhan lingkungan sekitar. Untuk itu, diperlukan pendekatan berbasis data, partisipasi masyarakat, serta evaluasi dan penyesuaian berkenlajutan guna memastikan manfaat yang maksimal bagi seluruh penduduk DKI Jakarta.

K-Means clustering merupakan metode yang sangat efisien untuk segmentasi data karena kemampuannya dalam mengelompokan data menjadi beberapa *cluster* yang homogen berdasarkan karakteristik tertentu ( L. Y. Hutabarat, 2021). Proses kerjanya melibatkan pemilihan sejumlah *cluster* awal (k), di mana data kemudian di tempatkan ke dalam *cluster* yang paling dekat berdasarkan jarak dari titik pusat (Centroid) *cluster* tersebut. Semua data akan ditempatkan dalam cluster, centroid dihitung ulang, dan diproses ini diulang hingga posisi centroid stabil dan perubahan dalam pengelompokan data minimal. Konsep ini memungkinkan analisis yang mendalam dengan memanfaatkan fitur karakteristik data untuk mengidentifikasi pola dan hubungan yang mungkin tidak terlihat secara langsung. K-Means clustering secara efisien mengelompokan dataset besar ke dalam kelompok-kelompok yang berbeda dengan cara meminimalkan variasi dalam *cluster* dan memaksimalkan jarak antar *cluster* (Parsa & Javidan, 2022). Dengan pendekatan ini, pemerintah dapat memperoleh wawasan yang lebih mendalam mengenai pola distribusi sosial ekonomi di DKI Jakarta, serta dapat merancang kebijakan yang lebih efektif dan tepat sasaran.

Berdasarkan permasalahan dan uraian diatas ,maka dalam penelitian ini penulis ingin mengangkat tema pemetaan kepadatan penduduk ke dalam beberapa kelompok (*cluster)*berdasarkan jumlah penduduk setiap wilayah DKI Jakarta, menggunakan K-Means clustering sebagai penulisan ilmiah.

## **1.2 Ruang Lingkup**

Berdasarkan latar belakang, ada beberapa batasan masalah dalam penelitian ini antara lain :

1. Data penduduk provinsi DKI Jakarta diambil dari website resmi satudata.jakarta.go.id
2. Pengelompokan data menggunakan metode K-Means Clustering.
3. Bahasa pemrograman yang digunakan untuk menerapkan metode tersebut adalah Python.

## **1.3 Tujuan Penelitian**

Tujuan dari penelitian ini adalah pemetaan penduduk provinsi DKI Jakarta menggunakan K Means Clustering untuk mengelompokkan wilayah DKI Jakarta berdasarkan kepadatan penduduk, dan membantu pemerintah untuk merancang program yang lebih efektif.

## **1.4 Metode Penelitian**

Metode penelitian yang digunakan pada penelitian ini menggunakan model standarisasi data mining, yaitu Knowledge Discovery in Database (KDD). Berikut adalah penjelasan tahapannya :

1. Data Selection, pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam Knowledge Discovery in Database (KDD) dimulai. Data hasil seleksi yang akan digunakan untuk proses data mining, disimpan dalam suatu berkas terpisah dari basis data operasional.
2. Data Cleaning / Pre-processing, sebelum proses data mining dapat dilaksanakan, perlu dilakukan proses cleaning pada data yang menjadi fokus Knowledge Discovery in Database (KDD). Proses cleaning mencakup antara lain membuang duplikasi data, memerikasa data yang tidak konsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak. Proses lainnya adalah *encrichment*, yaitu proses “memperkaya” data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk Knowledge Discovery in Database (KDD), seperti data atau informasi eksternal lainnya yang diperlukan.
3. Transformation Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses data mining. Proses coding dalam Knowledge Discovery in Database (KDD) merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.
4. Data Mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik-teknik, metode-metode, atau algoritma yang tepat sangat bergantung pada tujuan dan proses Knowledge Discovery in Database (KDD) secara keselurruhan.
5. Interpretation / Evaluation, pola informasi yang dihasilkan dari proses data mining perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses Knowledge Discovery in Database (KDD) yang disebut interpretation. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

Pada pembuatan program untuk menerapkan proses Clustering K-Means menggunakan perangkat lunak Microsoft Excel, Google Colab (IDE), Microsoft Windows 10 dan Web Browser.

## **1.5 Sistematika Penulisan Ilmiah**

Sistematika penulisan yang disusun penulis dalam penulisan ilmiah ini terdiri atas empat bab. Bab pertama adalah pendahuluan. Bab ini berisi mengenai latar belakang masalah, rumusan masalah, Batasan masalah, tujuan penelitian, metode penelitian, dan sistematika penulisan. Bab 2 merupakan Tinjauan Pustaka. Bab ini berisi teori yang mendukung penulisan yang meliputi teori-teori pendukung penelitian dan medianya, seperti data mining, clustering, algoritma K-Means, dan Bahasa pemrograman Python. Bab 3 adalah pembahasan, pada bab ini membahas mengenai penjelasan tahap penelitian yang dilakukan dimulai dari Gambaran umum pembuatan program, pengumpulan data dan menjelaskan tentang metode K-Means dan Langkah-langkahnya, serta pembahasan mengenai tahapan dalam pembuatan yang disertai contoh dan potongan program. Bab 4 adalah penutup, berisi Kesimpulan berdasarkan dari bab-bab yang ada sebelumnya beserta saran yang bermanfaat untuk semua pihak.

# **TINJAUAN PUSTAKA**

## **2.1 Data Mining**

Data mining merupakan serangkaian proses untuk menggali nilai tambah berupa informasi yang selama ini tidak diketahui secara manual dari suatu basis data. Data mining mulai ada sejak tahun 1990-an sebagai cara yang benar dan tepat untuk mengambil pola dan informasi yang digunakan untuk menemukan hubungan antara data untuk melakukan pengelompokkan ke dalam satu atau lebih *cluster* sehingga objek-objek yang berada dalam satu *cluster* akan mempunyai kesamaan yang tinggi antara satu dengan yang lainnya. Data mining merupakan bagian dari proses penemuan pengetahuan dari basis data *Knowledge Discovery in Database* (Alkhairi & Winarto, 2019).

Data mining dibagi menjadi beberapa kelompok berdasarkan dengan tugas yang dapat dilakukan (Rosmini et al., 2018), yaitu :

1. Deskripsi

Para peneliti dan analisis biasanya mencoba menemukan cara untuk menggambarkan pola dan trend yang tersembunyi dalam bentuk data.

1. Estimasi

Estimasi memiliki kemiripan dengan klasifikasi, kecuali variabel target estimasi lebih kearah numerik dari pada kearah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Pada peninjauan berikutnya, dilakukan estimasi nilai variabel target yang dibuat berdasarkan hasil perhitungan dan analisis terhadap nilai variabel prediksi, sehingga dapat memberikan Gambaran yang lebih akurat mengenai kecenderungan data.

1. Prediksi

Prediksi memiliki kemiripan dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil aka nada di masa yang akan datang. Beberapa algoritma dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

1. Klasifikasi

Dalam klasifikasi variabel, tujuan bersifat kategori, sebagai contoh pengklasifikasian persediaan dalam tig akelas, yaitu persedian tinggi, persediaan sedang, dan persediaan rendah.

1. *Clustering*

*Clustering* merupakan teknik pengelompokkan *record* data, pengamatan atau kasus dalam kelas yang memiliki kemiripan. *Cluster* adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidak miripan dengan *record* lain dalam *cluster.*

1. Asosiasi

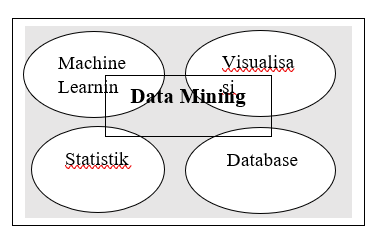
Mengidentifikasi hubungan antara berbagai peristiwa yang terjadi pada satu waktu. Dalam dunia bisnis lebih umum disebut sebagai analisiscitas keranjang belanja.

## **2.2 Pengenalan Pola Data Mining dan Machine Learning**

Pengenalan pola adalah suatu disiplin ilmu yang mempelajari cara-cara mengklasifikasikan objek ke beberapa kelas atau kategori dan mengenali kecenderungan data. Tergantu pada aplikasi, objek-objek ini bisa berupa pasien, mahasiswa, pemohon kredit, *image* atau *signal* atau pengukuran lain yang perlu diklasifikasikan atau dicari fungsi regresinya. Data mining, sering juga disebut Knowledge Discovery in Database (KDD), yaitu kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar. keluaran dari data mining ini bisa dipakai untuk memperbaiki pengembalian keputusan di masa depan.

Machine learning adalah pembelajaran mesin yang sangat membantu dalam menyelesaikan masalah, membuat mudah dalam mengerjakan suatu pekerjaan (Telaumbanua, F. D., & Dharma, A., 2019). Machine learning merupakan cabang ilmu bagian dari kecerdasan buatan (*Artificial Intelligence),* dengan pemrograman untuk memungkinkan komputer menjadi cerdas berperilaku seperti manusia, dan dapat meningkatkan pemahamannya melalui pengalaman secara otomatis. Machine learning memiliki fokus pada pengembangan sistem yang mampu belajar sendiri untuk memutuskan sesuatu tanpa harus berulangkali deprogram oleh manusia (Retnoningsih, E., & Pramudita, R., 2020).

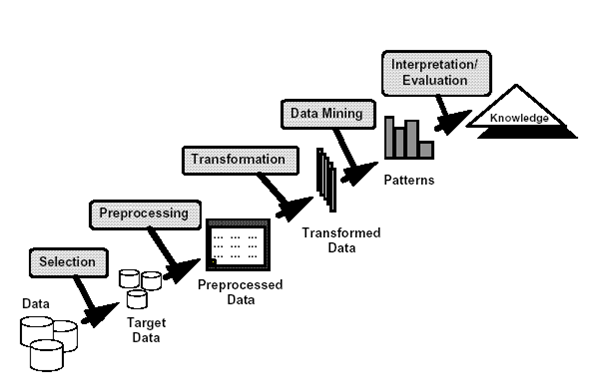
Machine learning merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) dan ilmu komputer yang berfokus pada penggunaan data dan algoritma untuk meniru cara manusia belajar dan secara bertahap dapat meningkatkan akurasinya. Semakin bagus algoritma machine learning yang digunakan maka akan semakin baik pula keputusan yang dihasilkan dan di dapatkan (Ahmad, A. 2017) . Hubungan ilmu data mining dan visualisasi ditunjjukan pada gambar 2.1.

****

Gambar 2. 1 Hubungan ilmu data mining dan visualisasi data

### **2.2.1Tahap-tahap data mining**

Istilah Knowledge Discovery in Database (KDD) dan data mining seringkali digunakan secara bergantian untuk menjelaskan proses penggalian informasi yang masih tersembunyi dalam suatu basis data yang besar. sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi saling berkaitan satu sama lain, dan salah satu tahapan dalam keseluruhan proses KDD adalah data mining. Proses KDD secara garis besar dapat digambarkan pada gambar 2.2.



Gambar 2. 2 Proses tahapan KDD

1. *Data Selection*

Pemilihan atau seleksi data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam Knowledge Discovery in Database (KDD) dimulai. Data hasil seleksi yang akan digunakan untuk proses data mining, disimpan dalam suatu berkas terpisah dari basis data operasional

1. *Data Cleaning* (*Pre-processing)*

Proses data mining dapat dilaksanakan, setelah dilakukan proses cleaning pada data yang menjadi fokus Knowledge Discovery in Database(KDD).Proses cleaning mencakup antara lain membuang duplikasi data, memeriksa data yang tidak konsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak. Juga dilakukan proses *encrichment,* yaitu proses memperkaya data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk Knowledge Discovery in Database (KDD), seperti data atau informasi eksternal lainnya yang diperlukan.

1. *Data Transformation*

Coding adalah proses transformasi data yang telah dipilih, sehingga data tersebut sesuai untuk proses data mining. Proses coding dalam Knowledge Discovery in Database (KDD) merupakan proses kreatif dan sangat bergantung pada jenis atau pola informasi yang akan dicari dalam basis data. Data dan atribut yang digunakan diambil dari database untuk dianalisis, dan selanjutnya data tersebut akan diubah menjadi bentuk yang tepat untuk *di-mining.*

1. *Data Mining*

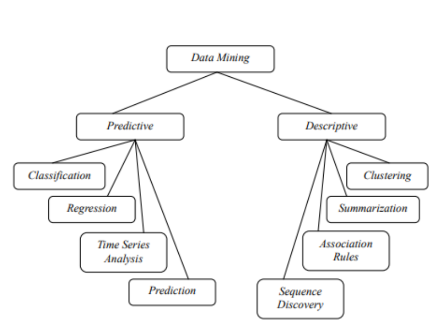
Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik-teknik, metode-metode, atau algoritma dalam data mining sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses Knowledge Discovery in Database (KDD) secara keseluruhan.

1. *Interpretation / Evaluation*

Pola informasi yang dihasilkan dari proses data mining perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses Knowledge Discovery in Database (KDD) yang disebut *interpretation.* Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

### **2.2.2Metode Data Mining**

Pada umumnya metode data mining dikelompokan ke dalam dua kategori, yaitu deskriptif dan prediktif. Metode deskriptif untuk mencari pola yang dapat dimengerti oleh manusia yang menjelaskan karakteristik dari data. Metode *clustering* menggunakan ciri-ciri tertentu dari data untuk melakukan pengelompokan data. Metode-metode dalam data mining terdapat pada gambar 2.3.



Gambar 2. 3 Metode data mining

Terdapat beberapa teknik yang digunakan dalam data mining, yaitu :

1. *Classification* / Klasifikasi

Klasifikasi adalah teknik yang paling umum diterapkan pada data mining pendekatan ini sering menggunakan keputusan pohon (*decision tree)* atau *neural network* berbasis algoritma klasifikasi. Proses klasifikasi data melibatkan learning dan klasifikasi. Dalam belajara (*learning)* data pelatihan *(training)* dianalisis dengan algoritma klasifikasi. Dalam klasifikasi pengujian data dilakukan dengan menggunakan perkiraan akurasi dari aturan klasifikasi. Jika akurasi bisa diterima, maka aturan dapat diterapkan untuk data baru. Salah satu contoh yang mudah dan popular adalah dengan *decision tree* yaitu salah satu metode klasifikasi yang popular karena mudah untuk diinterpretasi. *Decision tree*  adalah model prediksi menggunakan struktur pohon atau struktur berhirarki. *Decision tree* adalah struktur *flowchart* yang mempunyai *tree* atau pohon, dimana setiap simpul *internal* menandakan suatu tes pada atribut, setiap cabang merepresentasikan hasil tes, dan simpul merepresentasikan kelas atau distribusi kelas. Alur pada *decision tree* ditelusuri dari simpul akar ke simpul daun yang memegang prediksi kelas untuk contoh tersebut. *Decision tree* mudah untuk dikonversikan ke aturan klasifikasi (*classification rules*).

1. *Clustering*

*Clustering*  bisa dikatakan sebagai identifikasi kelas objek yang memiliki kemiripan. Dengan menggunakan teknik *clustering* dapat lebih lanjut untuk mengidentifikasi kepadatan dan jarak daerah dalam objek ruang dan dapat menemukan secara keseluruhan pola distribusi data korelasi antara atribut. Pendekatan klasifikasi secara efektif juga dapat digunakan untuk membedakanakelompok atau kelas objek.

1. *Predication*

*Predication* adalah Teknik regresi dapat yang disesuaikan untuk prediksi. Analisis regresi dapat digunakan untuk model hubungan antara satu atau lebih *independent variables* dan *dependent variables.* Dalam data mining *independent* variabel adalah atribut-atribut yang sudah dikenal dan respon variabel apa yang diinginkan untuk diprediksi, akan tetapi banyak masalah di dunia nyata bukan prediksi yang mudah. Teknik yang kompleks seperti *logistic regression,decision trees* atau pohon keputusan, *neural nets* atau jaringan syaraf, mungkin akan diperlukan untuk memprediksi nilai. Model yang berjenis sama sering dapat digunakan untuk regresi dan klasifikasi. Misalnya, CART (C*lassification and Regression Trees*) yaitu algoritma pohon Keputusan yang dapat digunakan untuk membangun kedua pohon klasifikasi dan pohon regresi. Jaringan saraf juga dapat menciptakan kedua model klasifikasi dan regresi.

1. *Association rule*

*Association rule* digunakan untuk mengenali kelakuan dari kejadian-kejadian khusus atau proses dimana *link* asosiasi muncul pada setiap kejadian. Contoh dari aturan asosiatif dari Analisa pembelian di suatu pasar swalayan adalah bisa diketahui berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu. Dengan pengetahuan tersebut pemilik pasar swalayan dapat mengatur penempatan barangnya atau merancang kampanye pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu. Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua *parameter, support* yaitu presentase kombinasi atribut tersebut dalam basis data dan *confidence* yaitu kuatnya hubungan antar atribut dalam aturan asosiatif. Motivasi awal pencairan *association rule* berasal dari keinginan untuk menganalisa data transaksi *supermarket,* ditinjau dari perilaku *customer* dalam membeli produk. *Association rule* ini menjelaskan seberapa sering suatu produk dibeli secara bersamaan. Sebagai contoh, *association rule “*Roti *=>* Selai *(75%)”* ini berati bahwa dari 75% dari pelanggan yang membeli roti juga membeli selai, jadi jika seseorang pelanggan membeli roti, memiliki kemungkinan yang tinggi bahwa juga akan membeli selai. Dalam satu *association rule* X => Y, X disebut dengan *antecendent* dan Y disebut dengan *consequent rule.*

1. *Neural network*/Jaringan Saraf

Jaringan saraf adalah seperangkat unitv penghubung *input & output* Dimana setiap koneksinya memiliki bobot. Selama fase *learning,* jaringan belajar dengan menyesuaikan bobot sehingga dapat memprediksi kelas yang benar label dari setiap input. Jaringan saraf memiliki kemampuan yang luar biasa untuk memperoleh arti dari data yang rumit atau tidak tepat dan dapat digunakan untuk mengambil pola-pola serta mendeteksi tren yang sangat komplek untuk diperhatikan baik oleh manusia atau teknik komputer. Jaringan saraf sangat baik untuk mengidentifikasi pola atau tren pada data dan sangat cocok untuk melakukan prediksi serta memprediksi kebutuhan.

1. *Decision trees*

Pohon Keputusan adalah struktur *tree-shaped* yang mewakili set keputusan. Keputusan ini menghasilkan aturan untuk klasifikasi dsebuah kumpulan data. Metode pohon keputusan diantaranya yaitu *classification* and *regression trees* (CART) dan *Chi Square Automatic Interaction Detection* (CHAID).

1. *Nearest Neighbor Method*

Teknik yang mengklasifikasikan setiap record dalam sebuah kumpulan data berdasarkan kombinasi suatu kelas k *record* yang sama dalam sebuah kumpulan data histori (Dimana k lebih besar atau sama dengan 1). Terkadang disebut juga dengan teknik *K-Nearest Neighbor.*

## **2.3 Pengklasteran (*Clustering*)**

*Clustering* adalah teknik untuk membedakan kumpulan data menjadi banyak kelompok dilihat dari kecocokan yang diinginkan. *Clustering* merupakan suatu teknik dalam bidang data mining yang bertujuan untuk mengelompokan data ke dalam *cluster* atau kelompok berdasarkan kesamaan karakteristik. Proses *clustering* melibatkan pengelompokan data berdasarkan jarak terdekat dengan objek lain dalam kumpulan data dan data tersebut dikelompokan secara acak, metode pengelompokan berbeda yang dapat diterapkan pada kumpulan data yang besar (S. Ika Murpratiwi et al, 2021).

Pengelompokan diperlukan karena data yang belum diolah sangat besar dan tidak mudah untuk dilakukan analisis maupun dipelajari. Tujuan pengelompokan dalam scenario ini adalah untuk lebih memahami data dan menganalisis kualitas data. Analisis ­*cluster* merupakan pengelompokan objek-objek data hanya berdasarkan pada informasi yang terdapat pada data, yang menjelaskan objek dan relasinya ( Javed Mehedi et al, 2020).

*Clustering* merupakan contoh dari pengelompokan tanpa arahan (*unsupervised*). Pengelopokan merujuk kepada prosedur yang menetapkan objek data set kelas. *Unsupervised* berati bahwa pengelompokan tidak tergantung pada standar kelas dan pelatihan atau training.

1. *Supervised Learning*

Teknik *supervised learning* merupakan teknik yang bisa diterapkan pada pembelajaran mesin yang bisa menerima informasi yang sudah ada pada data dengan memberikan label tertentu.. diharapka teknik ini bisa memberikan target terhadap output yang dilakukan dengan membandingkan pengalaman belajar di masa lalu.

1. *Unsupervised Learning*

Teknik *unsupervised learning* merupakan teknik yang bisa diterapkan pada machine learning yang digunakan pada data yang tidak memiliki informasi yang bisa diterapkan secara langsung. Teknik ini diharapkan dapat membantu menemukan struktur atau pola tersembunyi pada data yang tidak memiliki label.

## **2.4 Analisis Karakteristik Metode Clustering**

Metode clustering dapat dijalankan dengan adanya beberapa tahapan yang penting berikut tahapannya :

1. Menetapkan ukuran jarak antar data. Mengukur kesamaan antara objek sesuai prinsip dasar *cluster* yang mengelompokan objek yang mempunyai kemiripan, maka proses pertama adalah mengukur seberapa jauh adanya kesamaan objek. Pengukuran jarak yang popular adalah metode *Euclidean distance.* Pada dasarnya car aini akan memasukan sebuah data ke dalam *cluster* tertentu dengan mengukur jarak data tersebut ke pusat cluster.
2. Melakukan proses standarisasi data apabila diperlukan.
3. Melakukan pengklasteran. Proses inti clustering adalah pengelompokan data, yang biasa dilakukan dengan dua metode yaitu :
4. Metode Hierarki

Metode ini memulai pengelompokan dengan dua atau lebih objek yang mempunyai kesamaan paling dekat. Proses diteruskan ke objek lain yang mempunyai kedekatan kedua. Demikian seterusnya hingga *cluster* akan membentuk semacam pohon Dimana ada hierarki (tingkatan yang jelas) antara objek. Dendogram biasanya digunkan untuk membantu memperjelas proses hierarki tersebut.

1. Metode Non Hirarki

Berbeda dengan metode hirarki, metode ini justru dimulai dengan menentukan terlebih dahulu jumlah cluster diketahui, baru proses *cluster* dilakukan tanpa mengikuti proses hirarki. Metode ini juga dengan *K-Means Cluster.*

1. Melakukan penamaan *cluster-cluster* yang terbentuk.
2. Melakukan validasi dan *prfiling cluster.* Adapun ciri-ciri *cluster*  adalah homogenitas (kesamaan) yang tinggi antara anggota dalam satu *cluster* (*within cluster*) dan heterogenitas (perbedaan) yang tinggi antara cluster yang satu dengan *cluster* lain (*between-cluster*). Analisis *cluster* memliki beberapa istilah penting, antara lain :
3. *Distances between cluster centeres,* yaitu jarak yang menunjukan bagaimana terpisahnya pasangan individu *cluster.*
4. Keanggotaan *cluster* (*cluster membership*), ialah keanggotaan yang menunjukan *cluster* untuk setiap objek yang menjadi anggotanya.
5. Pusat *cluster* (*cluster centers*), ialah titik awal di mulai pengelompokan dalam  *cluster non* hirarki.
6. Rata-rata lama *cluster* (*cluster centroid*), ialah nilai rata-rata variable dari semua objek atau observasi dalam *cluster* tertentu.
7. Jadwal aglomerasi (*agglomeration schudle*), ialah jadwal yang memberikan informasi tentang objek atau kasus yang dikelompokan pada setiap tahap pada suatu proses analisis *cluster* yang hirarki.

*Clustering* merupakan proses membuat pengelompokan sehingga semua anggota dari setiap partisi mempunyai persamaan berdasarkan matriks tertentu. Analisis *cluster* atau analisis kelompok merupakan teknik analisis data yang bertujuan untuk mengelompokan individua tau objek ke dalam beberapa kelompok yang memiliki sifat berbeda antar kelompok, sehingga individu atau objek yang terletak di dalam satu kelompok akan mempunyai sifat relative homogen. Tujuan analisis *cluster* adalah untuk mengelompokan objek-objek tersebut.

Analisis metode *cluster* memiliki beberapa kelebihan dan juga kekurangan sebagai berikut :

1. Kelebihan dari metode *cluster* adalah :
2. Dapat menggelompokan data observasi dalam jumlah yang besar dan variable yang relative banyak. Data yang direduksi dengan kelompok akan mudah dianalisis.
3. Dapat dipakai dalam skala data ordinal, interval dan rasio.
4. Kekurangan dari metode *cluster* adalah :
5. Pengelompokan bersifat subjektivitas peneliti karena hanya melihat dari gambar dendogram.
6. Untuk data heterogen antara objek penelitian yang satu dengan yang lain akan sulit bagi peneliti untuk menentukan jumlah kelompok yang dibentuk.
7. Metode-metode yang dipakai memberikan perbedaan yang signifikan, sehingga dalam perhitungan biasanya masing-masing metode dibandingakan.
8. Semakin besar observasi, biasanya tingkat kesalahan akan semakin besar.

## **2.5 Algoritma K-Means**

*K-Means* merupakan suatu algoritma yang digunakan dalam pengelompokan secara partisi yang memisahkan data ke dalam kelompok yang berbeda-beda. Dalam algoritma K-Means, setiap data harus termasuk ke *cluster* tertentu dan bisa dimungkinkan bagi setiap data yang termasuk *cluster* tertentu pada suatu tahapan proses, pada tahapan berikutnya berpindah ke *cluster* yang lainnya (M. Wahyudi et al, 2020).

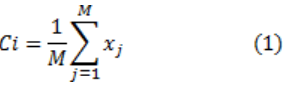
Algoritma ini mampu meminimalkan jarak antara data ke *cluster nya.* Pada dasarnya penggunaan algoritma ini dalam proses clustering tergantung pada data yang didapatkan dan konklusi yang ingin dicapai di akhir proses, sehingga dalam penggunaan algoritma *K-Means* terdapat aturan sebagai berikut :

1. Berapa jumlah *cluster* yang perlu dimasukan.
2. Hanya memiliki atribut bertipe *numeric.*

Pada dasarnya algoritma *K-Means* hanya mengambil sebgaian dari banyaknya komponen yang didapatkan untuk kemudian dijadikan pusat *cluster* awal, pada penetuan pusat *cluster* ini dipilih secara acak dari populasi data. Penentuan pusat cluster, setelah itu langkah selanjutnya dilakukan penentuan pusat *cluster,* algoritma *K-Means* akan menguji masing-masing dari setiap komponen dalam populasi data tersebut dan menandai komponen tersebut ke dalam salah satu pusat *cluster* yang telah didefiniskan sebelumnya tergantung dari jarak minimum antara komponen dengan tiap-tiap pusat *cluster,* selanjutnya posisi pusat *cluster* akan dihitung kembali sampai semua komponen data digolongkan ke dalam tiap-tiap *cluster* dan terkahir akan terbentuk *cluster* baru.

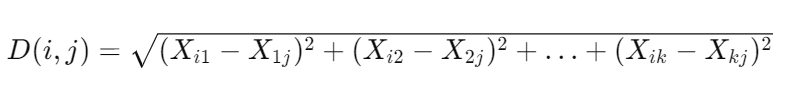
Algoritma *K-Means* pada dasarnya melakukan 2 proses yakni proses pendeteksian Lokasi pusat *cluster* dan proses pencarian anggota dari tiap-tiap *cluster.* Proses *clustering* dimulai dengan mengidentifikasi data yang akan di *cluster, Xij* (i=1,…,n; j=1,…,m). kemudian dihitung jarak antara setiap data dengan setiap puasat *cluster.* Untuk melakukan perhitungan jarak data ke-1 (xi) pada pusat *cluster* ke-k (ck) diberi nama (dik), dapat diguanakan formula *Euclidean.* Suatu data akan menjadi anggota dari *cluster* ke-k apabila jarak data tersebut ke pusat *cluster-k* bernilai paling kecil jika dibandingkan dengan jarak ke pusat *cluster* lain. Proses dasar algoritma *K-Means* dapat dilihat dibawah ini:

1. Tentukan jumlah *cluster* k
2. Alokasikan data ke dalam kelompok secara acak.
3. Hitung pusat kelompok (*centroid/rata-rata*) dari data yang ada di masing-masing kelompok. Lokasi centroid setiap kelompok diambil dari rata-rata (mean) semua nilai data pada setiap fiturnya. Jika M menyatakan jumlah data dalam sebuah kelompok, I menyatakan fitur ke-I dalam sebuah kelompok, dan p menyatakan dimensi data, maka persamaan untuk menghitung centroid fitur ke-I digunakan persamaan 1. Persamaan 1 dilakukan sebanyak p dimensi dari i=1 sampai dengan i=p. persamaan ini dapat dilihat pada rumus 2.1.



Rumus 2. 1 Rumus menentukan centroid

1. Rumus 2.1 adalah rumus untuk menentukan alokasikan masing-masing data ke centroid atau rata-rata terdekat. Ada beberapa cara yang dapat dilakukan untuk mengukur jarak data ke pusat kelompok, diantaranya adalah menggunakan metode *Euclidean*. Pengukuran jarak pada ruang jarak (*distance space*) *Euclidean* dapat dicari menggunakan persamaan seperti pada rumus 2.2.



Rumus 2. 2 Rumus jarak Euclidean

Dimana :

D (i,j) = Jarak data ke i ke pusat *cluster* j

Xki = data ke i pada atribut data ke k

Xkj = Titik pusat ke j pada atribut ke k

Pada rumus 2.2 yaitu rumus untuk menghitung jarak *Euclidean,* pengalokasian Kembali data ke dalam masing-masing kelompok dalam metode K-Means didasarkan pada perbandingan jarak antara data centroid setiap kelompok yang ada. Data dialokasikan ulang secara tegas ke kelompok yang mempunyai centroid dengan jarak terdekat dari data tersebut.

1. Apabila masih ada data yang berpindah kelompok atau apabila ada perubahan nilai *centroid* di atas nilai ambang yang ditentukan, atau apabila perubahan nilai pada fungsi objektif yang digunakan masih di atas nilai ambang yang ditentukan, maka kembali ke langkah ke 3.

### **2.5.1 Perbandingan K-Means dengan algoritma clustering lainnya**

K-Means clustering dikenal dengan kesederhanaan dan kecepatan komputasinya, menggunakan centroid sebagai pusat cluster dan sangat efisien untuk dataset yang besar. Beberapa algoritma clustering lainnya seperti Fuzzy C-Means juga dapat memberikan fleksibilitas tambahan dengan membolehkan setiap titik data memiliki keanggotaan probalistik di beberapa cluster, sehingga cocok untuk data dengan cluster yang saling tumpeng tindih. Contoh algoritma clustering lainnya adalah K-Medoids, menggunakan titik data aktual sebagai pusat cluster (medoid), menawarkan ketahahan lebih terhadap outlier dan memungkinkan penggunaan metrik jarak yang beragam. Tabel 2.1 menyajikan perbandingan mendetail dari ketiga algoritma ini.

Tabel 2. 1 Perbandingan antara K-Means, Fuzzy C-Means, dan K-Medoids

|  |  |  |
| --- | --- | --- |
| **K-Means** | **Fuzzy C-Means**  **(FCM)** | **K-Medoids** |
| Memiliki tipe hard clustering, setiap data diassign ke satu cluster tertentu. | Memiliki tipe soft clustering(non-ekslusif), data dapat memiliki keanggotaan pada beberapa cluster. | Memiliki tipe hard clustering (eksklusif), setiap data diassign ke satu cluster tertentu. |
| K-Means memiliki proses yang cepat karena peritungan centroid yang sederhana | Fuzzy C-Means lebih lambar karena harus menghitung keanggotaan fuzzy. | K-Medoids memiliki proses yang lebih lambat karena harus mengevaluasi medoid terlebih dahulu. |
| Untuk skalabilitas K-Means memiliki skala yang tinggi, efisien untuk dataset yang besar dan kompleks. | Fuzzy C-Means memeiliki skalabilitas rendah atau kurang efisien pada dataset yang besar. | K-Medoids juga memiliki sklabilitas yang rendah sehingga tidak cocok untuk dataset yang besar. |
| Metrik pengukuran jarak K-Means menggunakan jarak Euclidean sebagai default. | Metrik pengukuran jarak pada fuzzy C-Means umumnya menggunakan jarak Euclidean. | Metrik pengukuran jarak yang digunakan pada K-Medoids menggunakan jarak Manhattan. |
| Pusat cluster pada K-Means menggunakan centroid (rata-rata), pusat cluster dihitung sebagai rata-rata titik data dalam cluster. | Pusat cluster pada fuzzy C-Means menggunakan Fuzzy Centroid (Derajat Keanggotaan), pusat cluster dihitung berdasarkan derajat keanggotaan fuzzy. | Pusat cluster pada K-Medoids yaitu Medoid (titik data actual), pusat cluster adalah titik data actual yang mewakili cluster. |
| Efisiensi memori yang digunakan oleh K-Means sangat efisien karena hanya menyimpan centroid saja. | Efisiensi memori yang digunakan oleh Fuzzy C-means kurang efisien karena harus menyimpan derajat keanggotaan fuzzy. | Efisiensi memori yang digunakan sedang yaitu hanya diperlukan untuk menyimpan medoid. |
| K-Means cocok diterapkan untuk dataset yang besar, dan memiliki struktur yang jelas. | Fuzzy C-Means baik digunakan untuk data yang clusternya tumpeng tindih. | K-Medoids baik digunakan untuk data dengan outlier yang signifikan. |

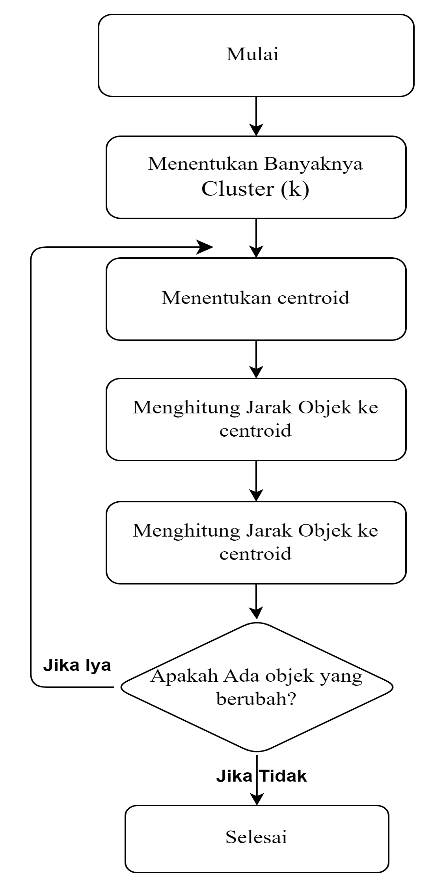
### **2.5.2 Kelebihan algoritma K-Means**

Algoritma K-Means merupakan metode clustering yang banyak digunakan karena berbagai kelebihannya. Salah satu keunggulannya adalah kesederhanaan dan efisiensinya. Algoritma K-Means tetap menjadi metode yang banyak digunakan karena kesederhanaan dan efisiensinya dalam menangani dataset yang besar (Hodge & Austin, 2023). Kemudahan implementasi dan efisiensi komputasi menjadikan pilihan praktis untuk berbagai tugas pengelompokan, selain itu K-Means juga dikenal memiliki skalabilitas yang baik, sehingga cocok untuk untuk aplikasi data skala besar dan bersifat linier terhadap jumlah titik data, yang memungkinkan penanganan dataset secara efektif. Keunggulan utama algoritma K-Means adalah kecepatannya, beroperasi dengan efisien dengan kompleksitas waktu yang umumnya dapat dikelola bahkan untuk dataset besar, yang penting dalam aplikasi waktu nyata (Smith Chen, 2024).

K-Means juga menawarkan efisiensi memori yang signifikan, karena K-Means hanya memerlukan penyimpanan posisi centroid dan penugasan cluster, sangat efisien dalam hal penggunaan memori, terutama saat menghadapi data dengan dimensi tinggi. K-Means efisien dalam penggunaan memori karena hanya menyimpan posisi centroid dan penugasan cluster, dan karakteristik ini sangat bermanfaat Ketika menghadapi data berdimensi tinggi (Jones Lee, 2023). Keunggulan ini menjadikan K-Means sebagai alat yang fleksibel dan dapat diandalkan untuk berbagai jenis analisis data, memastikan bahwa K-Means tetap relevan dan berguna dala berbagai aplikasi clustering.

## **2.6 Flowchart K-Means**

Flowchart *K-Means* menggambarkan urutan clustering dengan metode *K-Means* seperti yang terlihat pada gambar 2.4, merupakan flowchart yang berisi urutan proses dari mencari frekuensi kemunculan data , mencari jumlah cluster, menentukan centroid (titik pusat) awal, mencari jarak, mengelompokkan dokumen berdasarkan jarak terdekat dengan centroid serta proses mencari centroid baru. dalam *K-Means* pusat kelompok disebut dengan *centroid.* Perhitungan jarak ke pusat kelompok menggunakan rumus *Euclidean Distance* hingga ditentukan jarak terpendek antara setiap titik data ke centroid*.* Tahapan selanjutnya dilakukan pengelompokan objek berdasarkan jarak minimum terhadap pusat cluster. Pusat *cluster* tersebut kemudian untuk sementara dijadikan pusat cluster, atau centroid, mean, dan Jika masih terdapat objek yang harus dipindahkan ke *cluster* yang lain, maka proses diulang kembali, tetapi jika tidak maka proses selesai.



Gambar 2. 4 Flowchart K-Means

## **2.7 Bahasa Pemrograman Python**

Bahasa pemrograman Python telah menjadi salah satu bahasa pemrograman yang paling populer dan banyak digunakan di berbagai bidang, termasuk analisis data. Diciptakan pada tahun 1990 oleh Guido Van Rossum, python menawarkan fleksibilitas dan kemudahan bagi pengguna yang membuatnya menjadi pilihan utama bagi banyak ilmuwan data, peneliti dan pengembang perangkat lunak (Cahyadi,M.D.P.A., Tarjok, & Purwanto., 2021). Penggunaan Bahasa python meliputi analisis data, pengembangan web dan pengetesan perangkat lunak. Python menyediakan berbagai library, seperti NumPy untuk komputasi numerik dan Pandas untuk analisis data, yang memudahkan pengguna dalam menyelesaikan tugas-tugas tertentu dengan cepat dan efisien (Angelina M. T. I. Sambi Ua et al, 2023). Python juga merupakan Bahasa pemrograman dinamis yang mendukung perograman berbasis objek. Python didistribusikan dengan beberapa lisensi yang berbeda dari beberapa vaersi. Pada prinsipnya python dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial.

### **2.7.1 Fitur Python**

Hal yang membedakan python dengan Bahasa lain adalah dalam hal aturan penulisan kode program. Bahasa python juga mendukung hamper disemua system operasi, bahkan untuk system operasi linux, hamper semua distronya sudah menyertakan python didalamnya. Dengan kode yang simple dan mudah diimplementasikan, seorang programmer dapat lebih mengutamakan pengembangan aplikasi yang dibuat. Selain itu python merupakan salah satu produk yang open source juga multiplatform. Beberapa fitur yang dimiliki python adalah :

1. Memiliki Keputusan yang luas, dalam distribusi python telah disediakan modul-modul siap pakai untuk berbagai keperluan.
2. Memiliki tata bahasa yang jernih dan mudah dipelajari, dan memiliki aturan layout kode sumber yang memudahklan pengecekan, pembacaan Kembali dan penulisan ulang kode sumber yang berorientasi objek.
3. Memiliki system pengelolaan memori otomatis (*garbage collection*) seperti java modular, mudah dikembangkan dengan menciptakan modul-modul baru . modul tersebut dapat dibangun dengan Bahasa python maupun C atau C++
4. Memiliki fasilitas pengumpulan sampah otomatis, seperti halnya pada Bahasa pemrograman java, python memiliki fasilitas pengaturan penggunaan ingatan computer sehingga para programmer tidak perlu melakukan pengaturan ingatan computer secara langsung.

### **2.7.2 Library Python**

Library python memiliki beberapa library yang esensial untuk dikuasai oleh setiap data *scientist* yang ingin membuat model *Machine Learning,* yaitu

1. Library Numpy

Numpy adalah singkatan dari *Numerical Python.* secara sederhana, Numpy berisi kumpulan perhitungan matematika yang akan mempercepat *data Scientist* maupun *developer* Ketika ingin melakukan beberapa perhitungan matematis yang cukup rumit. Python juga berisi paket pemrosesan *array* untuk keperluan umum. Koleksi besar fungsi matematika dengan kompleksitas tinggi membuat Numpy kuat untuk memproses *array* dan matriks multidimensi yang besar. Numpy sangant berguna untuk menangani aljabar linear, transformasi *fourier,* dan bilangan acak. Dengan menggunakan NumPy, pemrosesan data numerik dapat dilakukan dengan lebih efisien dalam lingkungan bahasa pemrograman python ( Sardi et al, 2021). *Library* lain seperti *TensorFlow* menggunakan Numpy di backend untuk memanipulasi tensor, dengan Numpy, dapat menentukan tipe data *arbitrer* dan mudah berintegrasi dengan Sebagian besar database. Numpy juga dapat berfungsi sebagai wadah multidimensi yang efisien untuk data umum apa pun yang ada di tipe data apa pun. Numpy akan menjadi sangat esensial karena menjadi dasar dan landasan bagi beberap library lain seperti pandas, Matplotlib, Tensorflow, dan lain sebagainya.

1. Matplotlib

Matplotlib adalah Pustaka *plotting 2D* python yang menghasilkan gambar berkualitas publikasi dalam berbagai format *hard copy* dan lingkungan interaktif di seluruh *platform.* Matplotlib dapat digunakan dalam skrip python , shell python dan ipython, server aplikasi web dan enam *toolkit* antarmuka pengguna grafis. Matplotlib mencoba membuat hal yang mudah menjadi mudah dan hal yang sulit menjadi mungkin. Matplotlib dapat digunakan untuk membuat plot, histogram, spektrum daya, diagram batang, diagram kesalahan, diagram sebar hanya dengan beberapa baris kode.

1. Seaborn

Seaborn juga merupakan salah satu library untuk visualisasi data yang termasuk *high level.* Seabron adalah library untuk membuat grafik statistic dengan python. Seaborn dibangun diatas matplotlib dan terintegrasi erat dengan struktur data panda. Seaborn bertujuan untuk menjadikan visualisasi sebagai bagian sentral dari penjelajahan dan pemhaman data. Fungsi plotting berorintasi set data beroperasi pada kerangka data dan agregasi *array* yang berisi seluruh set data dan secara *internal* melakukan pemetaan semantic dan agregasi statistic yang diperlukan untuk menghasilkan plot informatif.

1. Pandas

Panda telah terbukti menjadi library python paling popular untuk analisis data dengan dukungan struktur data yang cepat, fleksibel dan ekspresif yang telah dikembangkan untuk data relasional dan berlabel. Pandas menjadi library yang esensial untuk menyelesaikan analisis data praktis dan nyata dengan python. Pandas sangat stabil dan menawarkan kinerja yang sangat optimal. Kode backend ditulis dalam C atau python. pandas juga merupakan fundamental dari proses data manipulation dan data mining. Bahkan, pandas juga dapat melakukan beberapa visualisasi data meski tidak seluas Matplotlib atau Seaborn. Terdapat dua tipe utama dari struktur data yang digunakan oleh pandas, antara lain adalah :

1. *Series* ( 1 dimensi ), berdiri di atas array.
2. *DataFrame* (2 dimensi), kumpulan dari series.

proses data manipulation biasanya dilakukan dengan memanipulasi kedua struktur data ini. Bisa jadi dengan memanipulasi array dan membuat data frame baru, atau melakukan proses manipulasi lainnya. Memanipulasi *series* dan *data frame* juga sangat berguna untuk melakukan visualisasi data. Menyiapkan DataFrame yang bersih menggunakan pandas merupakan langkah penting sebelum melakukan visualisasi data dengan matplotlib, dengan memiliki data yang terstruktur dan bebas dari kesalahan, proses visualisasi data akan menjadi lebih efektif hasilnya akan lebih akurat serta mudah dipahami.

1. Scikit-Learn

Scikit-learn adalah salah satu library machine learning yang paling popular. Scikit-learn mendukung banyak algoritma machine learning. Baik itu *Supervised Learning* maupun *Unsupervised Learning.* Di antara contoh model machine learning yang termasuk ke dalam library ini adalah seperti Linear Regression, Logistic Regression, Decision Tree Clasifier and Regresor, Random Forest Classifier and Regressor, K-Means clustering. Scikit-learn dibangun atas dua library dasar python yaitu Numpy dan SciPy. Scikit-learn menambahkan satu set algoritma untuk machine learning umum dan tugas penambangan data (data mining), termasuk pengelompokan, regresi, dan klasifikasi. Bahkan tugas-tugas seperti mengubah data, *Feature Selection* dan *Ensemble Methods* dapat diimplementasikan dalam beberapa baris saja. Untuk pemula dalam machine learning, scikit-learn sudah cukup sebagai alat untuk memulai, sebelum beralih ke implementasi algoritma yang lebih kompleks.

## **2.8 Pengujian Sillhoutte Score**

Pengujian *sillhoutte score* adalah proses evaluasi yang digunakan untuk mengukur seberapa baik suatu algoritma *clustering* mampu memisahkan data menjadi kelompok-kelompok yang berbeda. *Sillhoutte score* digunakan untuk mengukur seberapa dekat objek dalam *cluster* yang sama dan seberapa jauh objek tersebut dengan *cluster* lain, sehingga dapat menentukan seberapa baik *clustering* yang dilakukan (Purwanto, 2020). Metode ini melibatkan perhitungan *silhouette score* untuk setiap titik data, yang mencerminkan seberapa baik titik tersebut cocok dengan kelompoknya sendiri dibandingkan dengan kelompok lainnya. Pengujian *silhouette score* dilakukan dengan membagi data menjadi kelompok-kelompok dengan berbagai jumlah *cluster* yang berbeda dan membaandingkan nilai *silhouette score* untuk masing-masing pengelompokan. Perhitungan *silhouette score* dilakukan dengan rumus (b-a) / max (a,b), Dimana ‘a’ adalah jarak rata-rata data ke anggota *cluster* yang sama, dan ‘b’ adalah jarak rata-rata data ke *cluster* terdekat lainnya.

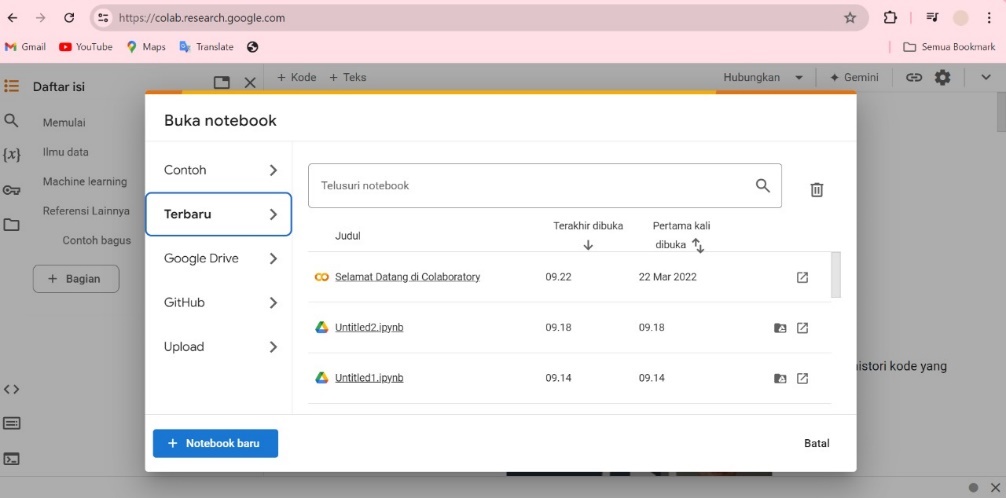
Nilai *silhouette score* yang tinggi menandakan bahwa *clustering* tersebut baik, sementara nilai yang rendah menunjukan bahwa ada penyebaran yang tidak homogen dalam satu atau lebih kelompok. Nilai yang dihasilkan berkisar antara -1 hingga 1, dengan skor yang lebih mendekati 1 menunjukan *clustering* yang lebih baik, sedangkan skor negating menunjukan potensi kesalahan dalam pengelompokan data. Oleh karena itu, pengujian *silhouette score* membantu dalam pemilihan jumlah *cluster* yang optimal untuk *clustering* suatu dataset. Setelah menghitung *silhouette score* untuk berbagai skenario *clustering* dengan jumlah *cluster* yang berbeda, langkah berikutnya dalam pengujian ini adalah memilih jumlah *cluster* yang memberikan nilai *silhouette score* tertinggi. Jumlah *cluster* ini merupakan pilihan yang optimal untuk membagi data menjadi kelompok-kelompok yang saling terpisah dengan baik. Pengujian *silhouette score* sangat berguna dalam menentukan jumlah *cluster* yang optimal, sehingga dapat meningkatkan akurasi dalam pengelompokan dat (Syahrir, 2019). Hasil pengujian *silhouette score* memberikan panduan yang berharga dalam proses pengelompokan data yang efektif dan informatif.

## **2.9 Google Colaboratory**

Google colab salah satu platform yang memiliki tujuan untuk mempermudah pekerjaan yang berkaitan dengan data science ( Guntara, R. G., 2023), selain itu *Google colab* bisa digunakan secara bersama-sama oleh pengembang aplikasi, sehingga sangat mendukung kebutuhan kolaborasi antar anggota tim. Google colab memang masih sangat awam bagi orang biasa karena hanya digunakan oleh orang yang membutuhkan coding environment, seperti para developer atau programmer. Google colab memiliki fitur kolaborasi yang memungkinkan para pengembang aplikasi dapat berkolaborasi antar tim dalam mengerjakan proyek yang cukup besar ( Setiadi, A. W. B. & Halim, S., 2022).

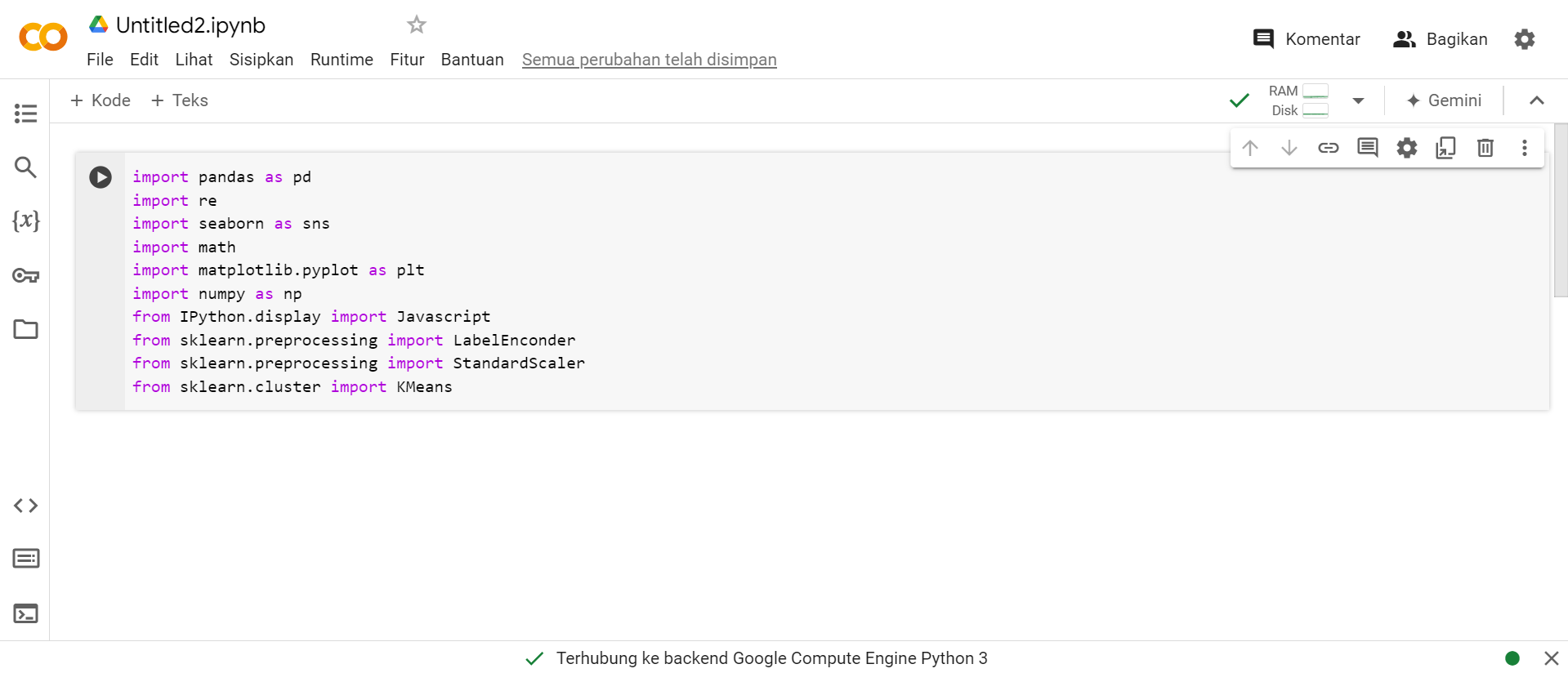
Colaboratory atau “colab” merupakan produk dari *Google Research*. Colab memungkinkan siapa saja menulis dan mengeksekusi kode Python arbitrer melalui browser, dan sangat cocok untuk machine learning, analisis data, serta Pendidikan. Secara lebih teknis, colab merupakan layanan notebook Jupyter yang dihosting dan dapat digunakan tanpa penyiapan, serta menyediakan akses gratis ke *resource* komputasi termasuk GPU. *Resource* colab tidak dijamin dan sifatnya terbatas, serta batas penggunaanya terkadang berfluktuasi. Hal ini diperlukan agar colab dapat menyediakan *resource* secara gratis. Pengguna yang ingin memiliki akses lebih andal ke *resource* yang lebih baik dapat menggunakan colab pro.

Colab pro merupakan langkah pertama yang Google ambil untuk melayani pengguna yang ingin melakukan lebih banyak hal di colab. Tujuan jangka panjang pihak Google adalah untuk terus menyediakan versi gratis colab. Saat yang bersamaan berkembang secara berkelanjutan untuk memenuhi kebutuhan pengguna Google. Hal pertama yang harus dilakukan setelah memasuki Google Colab adalah membuat notebook baru. Tampilan awal Google Colab terdapat pada gambar 2.5.



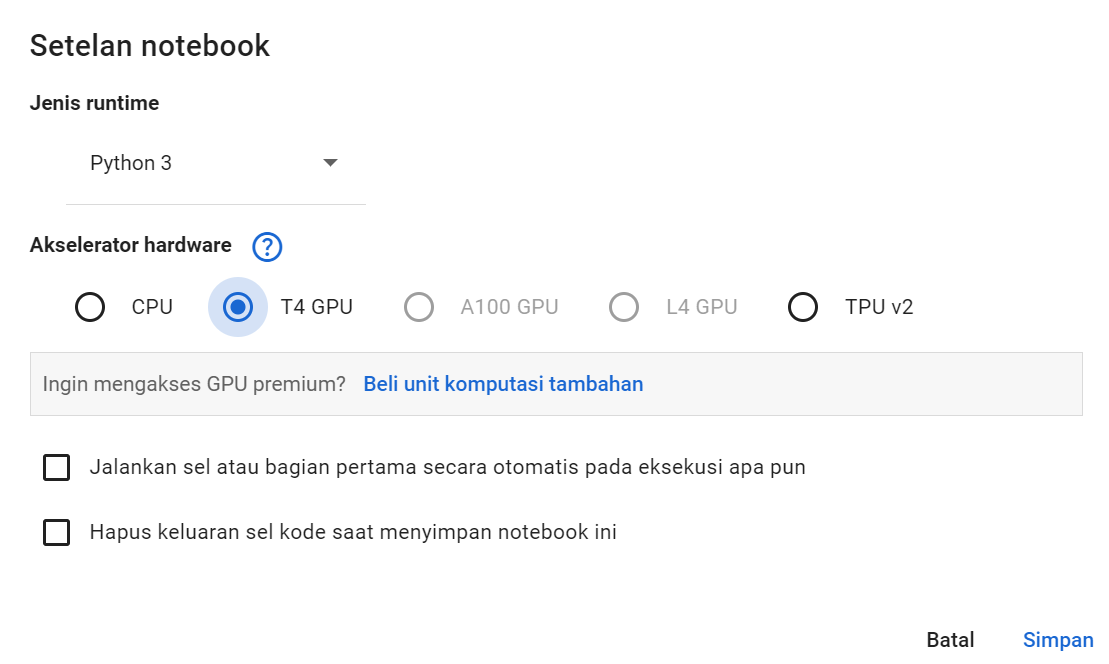
Gambar 2. 5 Tampilan awal Google Colab

Cara untuk membuat notebook baru, dengan cara klik new notebook lalu akan muncul halaman yang mirip dengan Jupyter Notebook, nantinya setiap notebook yang dibuat akan disimpan di Google Drive. Gambar 2.6 merupakan contoh input kode program pada Google Colab.



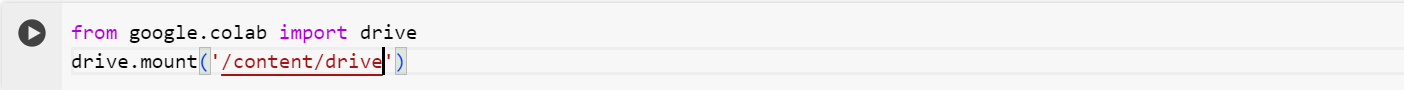
Gambar 2. 6 Contoh input kode program

Pada Google Colab terdapat pilihan untuk menjalankan program Python menggunakan GPU (atau bahkan TPU), di Google colab pilih “*Edit > setelan notebook”,* lalu pada bagian “*Hardware Accelerator”* pilih GPU. Gambar 2.7 menunjukan tampilan notebook Google Colab.



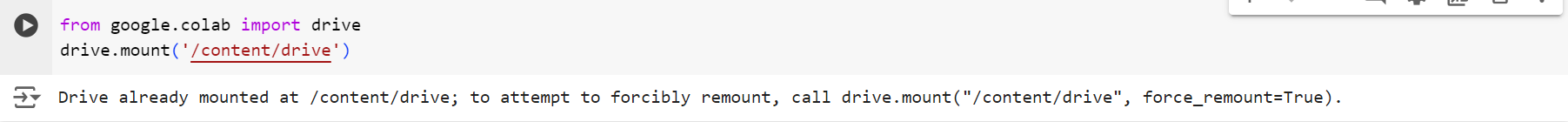
Gambar 2. 7 Pengaturan Google Colab

Menghubungkan dengan google drive, Google Colab akan mereset notebook beserta semua temporary maksimal 12 jam sekali dan disarankan akan lebih baik jika file yang akan digunakan atau dihasilkan tersimpan dengan rapih di Google Drive.



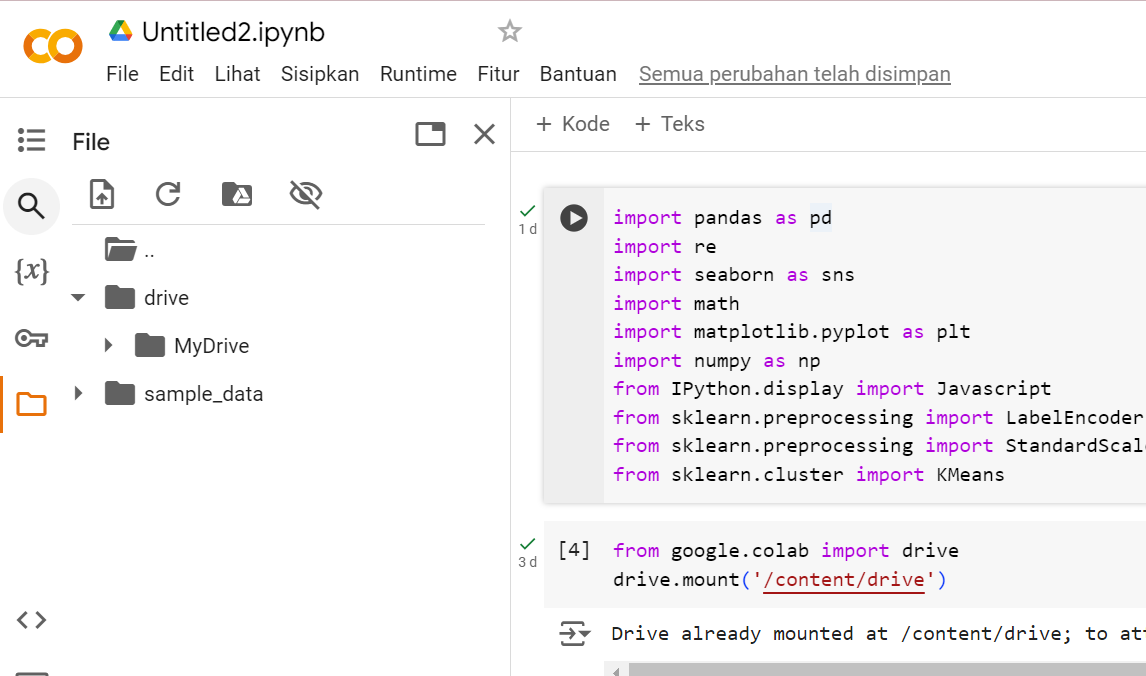
Gambar 2. 8 Input kode untuk koneksi drive

Jika input kode pada gambar 2.8 dijalankan, maka akan diberikan URL yang akan mengantarkan ke halaman permohonan akses Google Drive. Jika sudah klik izinkan, maka akan tampil output tersebut, seperti pada gambar 2.9



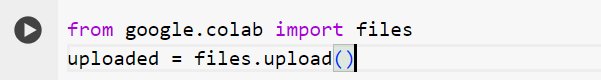
Gambar 2. 9 Tampilan berhasil mengakses drive

Jika sudah terhubung, maka akan tampak daftar file dibagian kiri notebook seperti pada gambar 2.10.



Gambar 2. 10 Daftar file dan folder

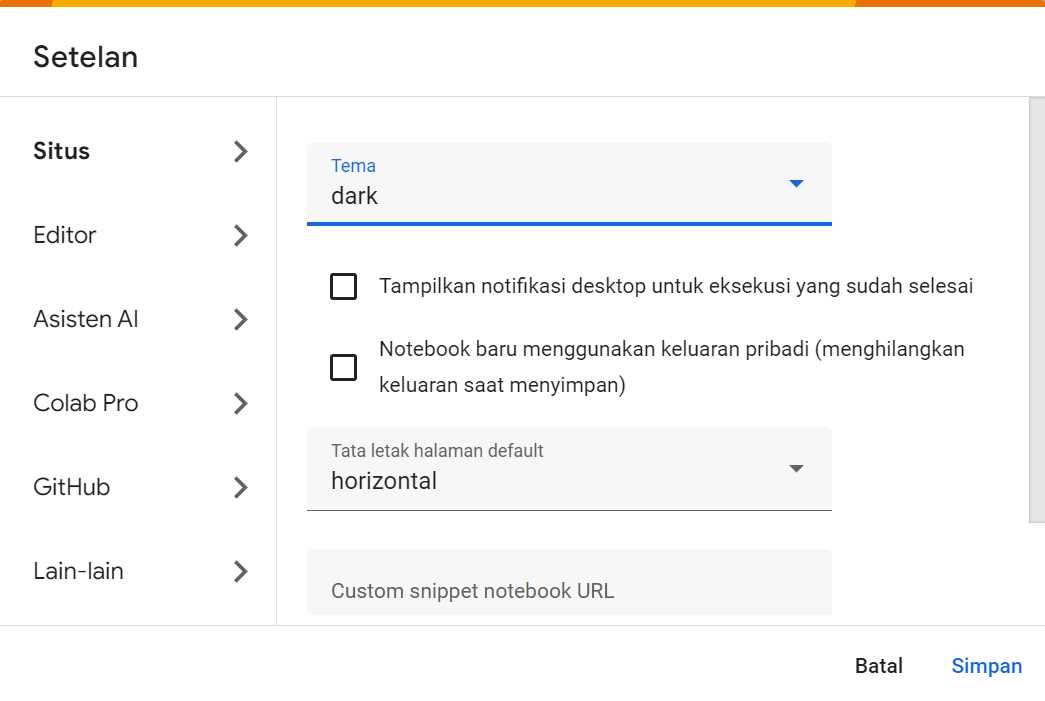
Cara untuk mengakses file-file tersebut, bisa dilakukan dengan mengarahkan proses load / save ke path. Meng-upload file ke colab dengan input kode : drive/My Drive/folder tujuan. Alternatif lain jika tidak ingin menghubungkan Google Colab k e Google Drive adalah dengan mengunggah langsung file yang diperlukan ke Colab. Colab menyediakan tempat penyimpanan file sementara yang akan di restart (dihapus) dalam rentang waktu tertentu. Cara untuk meng-upload file pada Google Colab bisa dilakukan dengan menjalankan input kode pada gambar 2.11.



Gambar 2. 11 Input kode untuk upload file

Jika perintah di atas dijalankan maka akan muncul kotak dialog untuk mengupload file. Perintah di atas cukup praktis untuk mengunggah file-file yang berukuran kecil (bukan dataset besar, lebih mudah diletakan di drive).

Google Colab juga menyediakan pilihan untuk mengubah tema notebook-nya menjadi gelap (*darkmode)*. Pengubahan tema dilakukan pada menu *tools > preferences > site.* Tampilan menu setting pada Google Colab dapat dilihat pada gambar 2.12.



Gambar 2. 12 Tampilan menu setalan Google Colab

Google Colab notebook digunakan secara luas oleh komunitas machine learning yang penerapannya meliputi beberapa hal seperti TensorFlow, mengembangkan dan melatih jaringan Neural, percobaan denga TPU, dan menyebarkan penelitian menggunakan AI. Beberapa kelebihan Google Colab antara lain tidak perlu konfigurasi, akses gratis ke GPU (*General Processing Units*), mudah untuk dibagikan, dan dapat impor langsung data ke dalam Colab Notebook melalui akun Google Drive, spreadsheet, GitHub, dan lain-lain.

## **2.10 Kepadatan Penduduk**

Tingkat kepadatan penduduk di Indonesia saat ini sulit diprediksi dengan cepat, hal ini dikarenakan adanya pergerakan penduduk yang tidak terkendali khususnya dari kawasan perdesaan ke perkotaan akibat tidak meratanya pembangunan ekonomi wilayah ( I. Indriani, D. Siregar & A. P. Windarto, 2022). Pergerakan ini dipengaruhi kondisi ekonomi penduduk yang menggantungkan hidupnya dari mencari pekerjaan di wilayah lain ( C. Adi Rahmat & Y. Novianto, 2021). Imbasnya peningkatan dan kepadatan penduduk suatu wilayah di Indonesia seringkali berubah dengan cepat serta berpotensi menimbulkan permasalahan baru, seperti meningkatkan angka kemiskinan, pengangguran dan bahkan tingkat kejahatan.

Kepadatan penduduk adalah perbandingan antara jumlah penduduk dengan luas wilayah yang dihuni .Kepadatan penduduk merupakan indikator dari pada tekanan penduduk di suatu daerah. Kepadatan di suatu daerah dibandingkan dengan luas tanah yang ditempati dinyatakan dengan banyaknya penduduk per kilometer persegi.Permasalahan kepadatan penduduk sangat mempengaruhi pengampu kebijakan dalam merencanakan dan menentukan arah pembangunan dimasa mendatang termasuk sulitnya pemerintah dan organisasi terkait untuk menghasilkan kebijakan dan strategi yang tepat untuk mengatasi pertumbuhan penduduk.

Berdasarkan para ahli tentang kepadatan penduduk maka dapat disimpulkan bahwa kepadatan penduduk merupakan suatu keadaan di mana semakin padat jumlah manusia pada suatu wilayah yang dihuni. Dalam hal ini luas wilayah tidak dapat mencukupi kebutuhan penduduk akan ruang di suatu pemukiman. Kepadatan penduduk yang tidak terkendali mengakibatkan dampak yang buruk terhadap lingkungan seperti semakin terbatasnya sumber daya pokok, tidak tercukupinya fasilitas sosial dan kesehatan, dan tidak tercukupinnya lapangan pekerjaan bagi tenaga kerja yang ada.

## **2.11 Satu Data Jakarta**

Satu data Jakarta adalah platform yang menyediakan data dalam format yang mudah dicari, diakses dan digunakan Kembali. Publik pengguna portal data terpadu yang akurat dan mutakhir. Portal ini akan terus menyajikan data dan informasi yang dibutuhkan oleh masyarakat dalam rangka penerapan *e-government,* pemenuhan hak publik, dan pewujudan tata Kelola pemerintahan yang transparan dan akuntabel.

Sesuai dengan Undang-Undang Nomor 14 Tahun 2008 tentang keterbukaan informasi publik, seluruh kumpulan data yang disajikan dalam portal data terpadu pemprov DKI Jakarta, satudata.jakarta.go.id, dapat dikategorikan sebagai domain public. Data yang tersaji tidak diperkenakan mengandung informasi terkait rahasia negara, rahasia pribadi, atau hal-hal lain yang telah diatur dalam undang-undang.

## **2.12 DKI Jakarta**

DKI Jakarta adalah ibu kota negara Indonesia dan menjadi pusat pemerintahan, ekonomi, serta kebudayaan. Terletak di pantai barat laut pulau jawa, Jakarta merupakan kota terbesar di Indonesia dengan populasi yang terus meningkat.Secara geografi, Jakarta memiliki luas wilayah sekitar 662,33 km2 termasuk daratan dan wilayah perairan di kepulauan seribu . Berdasarkan data Badan Pusat Statistik (BPS) pada tahun 2020, jumlah penduduk DKI Jakarta telah mencapai lebih dari 10,5 juta jiwa, dengan tingkat kepadatan tertinggi di Indonesia, sekitar 15.000 jiwa per kilometer persegi. Sebagai pusat ekonomi terbesar di Indonesia, Jakarta memiliki aktivitas ekonomi yang beragam seperti sektor jasa, perdagangan, industri, dan keungan membuat kota ini menjadi tujuan utama migrasi dari berbagai daerah. Laju pertumbuhan populasi yang cepat juga menimbulkan berbagai tantangan, terutama terkait dengan penyedian infrastruktur dan pelayanan publik.

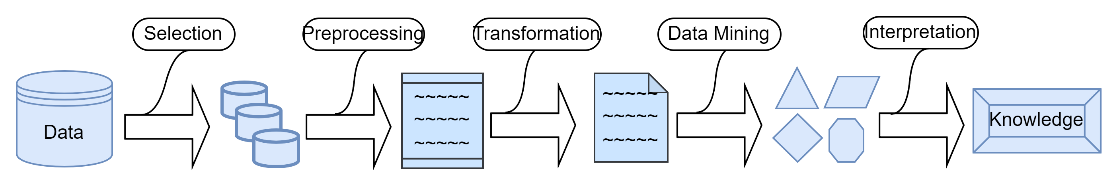
DKI Jakarta menghadapi berbagai masalah perkotaan seperti kemacetan lalu lintas, polusi udara yang tinggi, dan banjir. Pemerintah DKI Jakarta terus berupaya untuk meningkatkankualitas hidup warga DKI Jakarta melalui berbagai program pembangunan berkelanjutan, termasuk dalam mengatasi tantangan kepadatan penduduk yang semakin melonjak tinggi. Tingkat kepadatan penduduk DKI Jakrta menyebabkan beragam permasalahan sosial dan lingkungan, selain itu distribusi penduduk yang tidak merata antara wilayah pusat dan pinggiran kota menyebabkan masalah ketimpangan akses terhadap fasilitas umum dan pelayanan publik. Menurut kementrian Pekerjaan Umum dan Perumahan Rakyat (PUPR) pada tahun 2021, beberapa wilayah di Jakarta mengalami kekurangan ruang terbuka hijau, dan peningkatan jumlah pemukimah kumuh menjadi salah satu indikasi tekanan yang dihadapi kota ini dalam menyediakan perumahan yang layak.

# **PEMBAHASAN**

## **3.1 Gambaran Umum Penerapan *Clustering* Data Mining**

Perencanaan penulis adalah untuk mengembangkan sebuah penerapan data mining untuk pemetaan data kepadatan penduduk provinsi DKI Jakarta. Proses pengelompokan ini dikembangkan dengan aturan *clustering* menggunakan algoritma K-Means dan Bahasa pemrograman Python. Data yang digunakan untuk penulisan ini adalah data jumlah penduduk DKI Jakarta pada tahun 2018 yang diambil dari *website* satudata.jakarta.go.id. Hasil dari pemetaan ini dapat digunakan oleh pemerintah untuk merancang dan mengoptimalisasi program-program yang dibuat.

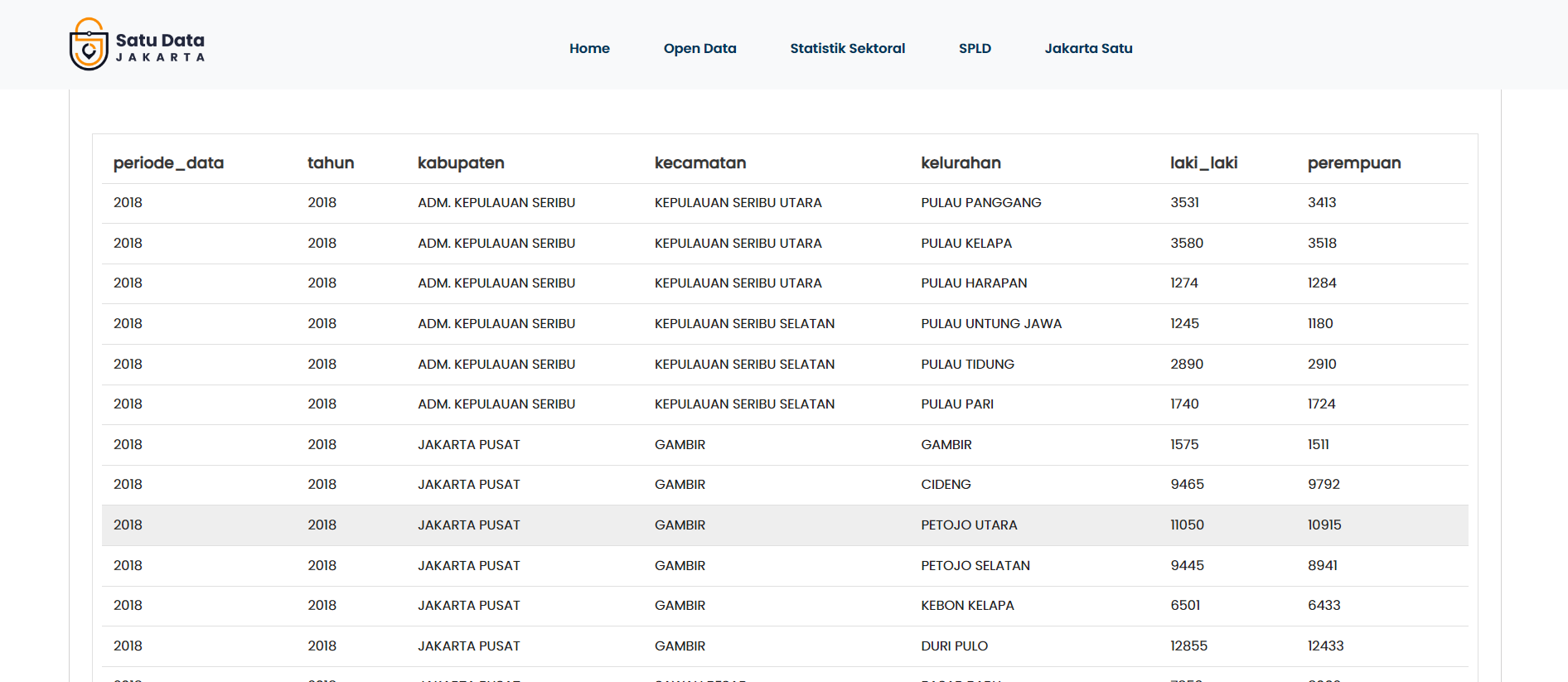
Pemetaan data jumlah penduduk DKI Jakarta ini dilakukan secara bertahap menggunakan metode KDD (*Knowledge Discovery in Database*). Tahapan yang dilalui dalam penulisan ini meliputi *data selection, pre-processing/cleaning, transformation, data mining,* dan *interpretation/evaluation* Gambaran tahap KDD (*Knowledge Database in Discovery*) terdapat pada gambar 3.1



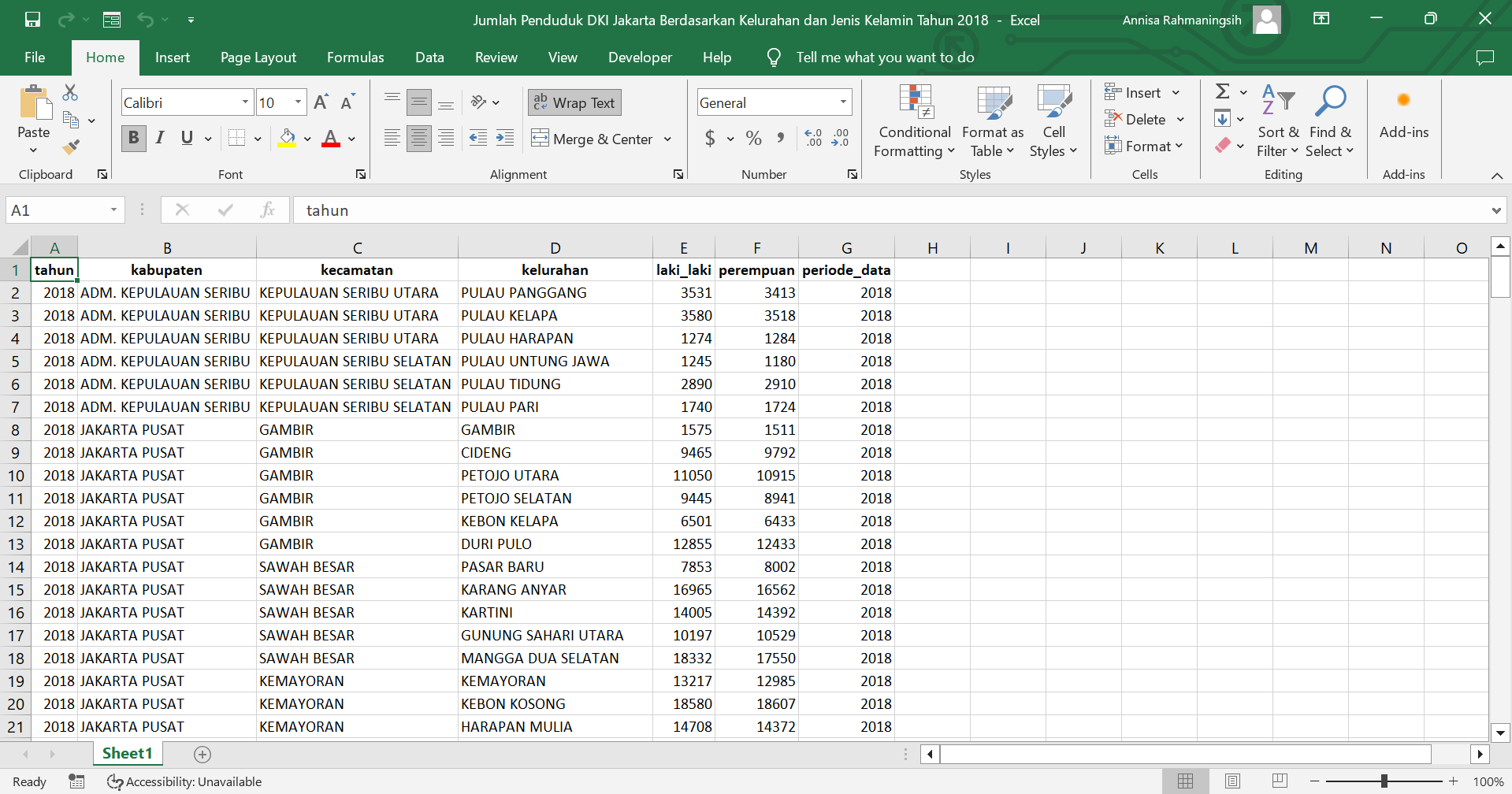
Gambar 3. 1 Tahapan penulisan menggunakan KDD

## **3.2 Data Selection**

*Data selection* adalah proses pemilihan data yang relevan dengan analisis dari basis data, penulis menggunakan data kepadatan penduduk DKI Jakarta pada tahun 2018. Data ini mencakup informasi tentang jumlah penduduk di berbagai wilayah DKI Jakarta. Pada tahapan ini dilakukan teknik perolehan sebuah pengurangan representasi dari data dan meminimalkan hilangnya informasi data. Hal ini meliputi metode pengurangan atribut dan komprensi data. Data yang digunakan diperoleh dari *website* satudata.jakarta.go.id, dapat dilihat pada gambar 3.2, selanjutnya ditampilkan informasi data pada Gambar 3.3.

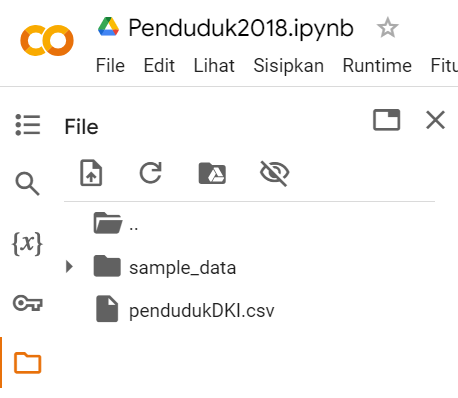


Gambar 3. 2 Data penduduk provinsi DKI Jakarta dari satudata.jakarta.go.id



Gambar 3. 3 Data yang sudah di download dengan format .CSV

Data dimasukan ke dalam sistem dengan menggunakan format .csv seperti pada gambar 3.3, kemudian disimpan kembali dalam format yang sama. Pemilihan format .csv ini didasarkan pada penggunaan library pandas, yang merupakan salah satu alat yang handal dan efisien untuk mengelola data. Library pandas menawarkan kestabilan yang sangat baik, serta kinerja yang optimal dalam menangani berbagai jenis data. Dengan menggunakan formta .csv dan library pandas bersamaan akan memberikan solusi yang efektif untuk mengelola data.



Gambar 3. 4 Mengunggah data .CSV ke Google Colab

Data.csv tersebut diunggah Google Colab seperti yang ditunjukan pada gambar 3.4 dengan cara memilih menu file dengan ikon folder lalu memilih unggah *session storage* yang dilambangkan dengan gambar kertas memiliki tanda panah ke atas. Pilih file yang akan diunggah ke session storage tersebut. Data yang ada pada database sering kali tidak semuanya terpakai, oleh karena itu data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Tabel 3.1 adalah tabel atribut apa saja yang akan digunkan dalam proses clustering K-Means data jumlah penduduk DKI Jakarta.

Tabel 3. 1 Atribut yang digunakan

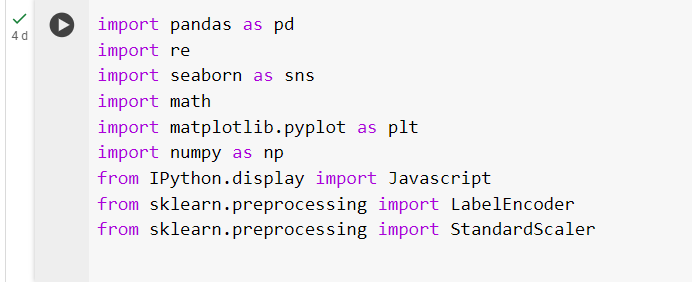
|  |  |
| --- | --- |
| **Aribut** | **Data yang digunakan** |
| Periode data | Tidak |
| Kabupaten | Ya |
| Kecamatan | Ya |
| Kelurahan | Ya |
| Jumlah laki-laki | Ya |
| Jumlah Perempuan | Ya |

## **3.3 Pre-processing/cleaning**

Proses *cleaning* mencakup antara lain membuang duplikasi data, memerikasa data yang tidak konsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak. Pembuatan program dilakukan dengan menggunakan media Google Colab, ada beberapa langkah untuk pembuatan programnya dari menyiapkan halaman Google Colab, , *import library* yang dibutuhkan, membaca data mentah dan menampilkannya, menentukan variabel *cluster* dan membuat *array-*nya*,* menstandarkan ukuran variabel, membuat fungsi K-Means, menentukan *cluster* dari data, menampilkan pusat dan hasil *cluster*, menambahkan kolom *cluster* pada tabel dan menampilkan hasil *clustering* data dan menghitung rata-ratanya, terakhir menampilkan grafik peta sebaran data yang sudah melewati proses *clustering*.

### **3.3.1 Import Library**

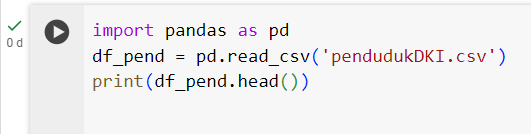
Dalam pembuatan program ini, terdapat beberapa package yang perlu diimport, *library-library* yang digunakan antara lain, ada library panda untuk membaca data dan analisis, library numpy untuk mempermudah pembuatan array dan pencarian *index* pada *array, library* matplotlib untuk memvisualisasikan data dengan grafik sebaran, library seaborn yang memiliki fungsi sama seperti library matplotlib. Gambar 3.5 adalah kode yang digunakan untuk mengimport *library* yang digunakan.



Gambar 3. 5 Kode Import Library

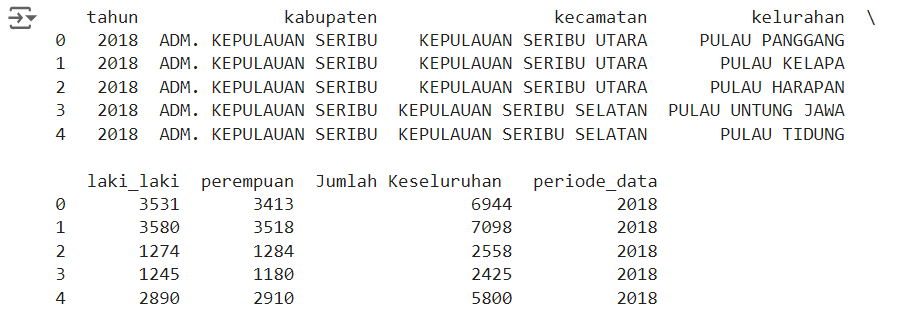
### **3.3.2 Data mentah**

Data yang sudah diunggah akan dibaca oleh program dan menampilkan data mentah tersebut dengan tujuan memeriksa kembali data yang sudah diunggah sehingga tidak terjadi kekeliruan data. Data mentah ini bisa digunakan untuk implementasi karena metode K-Means termasuk kedalam *unsupervised learning.* Data dibaca dengan menggunakan library pandas yang berfungsi untuk membaca file .csv, karena library ini efektif untuk menyelesaikan analisis data dan sangat stabil dengan kinerja yang sangat optimal. Berikut adalah input kode untuk *reading data* terdapat pada gambar 3.6.



Gambar 3. 6 Kode yang digunakan untuk menampilkan beberapa baris data

Dalam memasukan kode memuat data ke dalam *DataFrame* dan menggunakan `print(df\_pend.head())` untuk menampilkan lima baris pertama, yang memberikan gambaran awal tentang struktur data yang sudah ada. Tampilan awal dari dataset ini ditunjukan pada gambar 3.7 yang memperlihatkan kolom-kolom seperti kecamatan, kelurahan, jumlah penduduk laki, jumlah penduduk Perempuan, dan jumlah penduduk keseluruhan. Dataset ini mengandung informasi penting mengenai jumlah penduduk DKI Jakarta.



Gambar 3. 7 Tampilan data mentah .CSV

Tampilan dataset seperti pada gambar 3.7 dapat diketahui beberapa informasi penting tentang jumlah penduduk provinsi DKI Jakarta pada tahun 2018. Dataset ini memiliki kolom-kolom sebagai berikut :

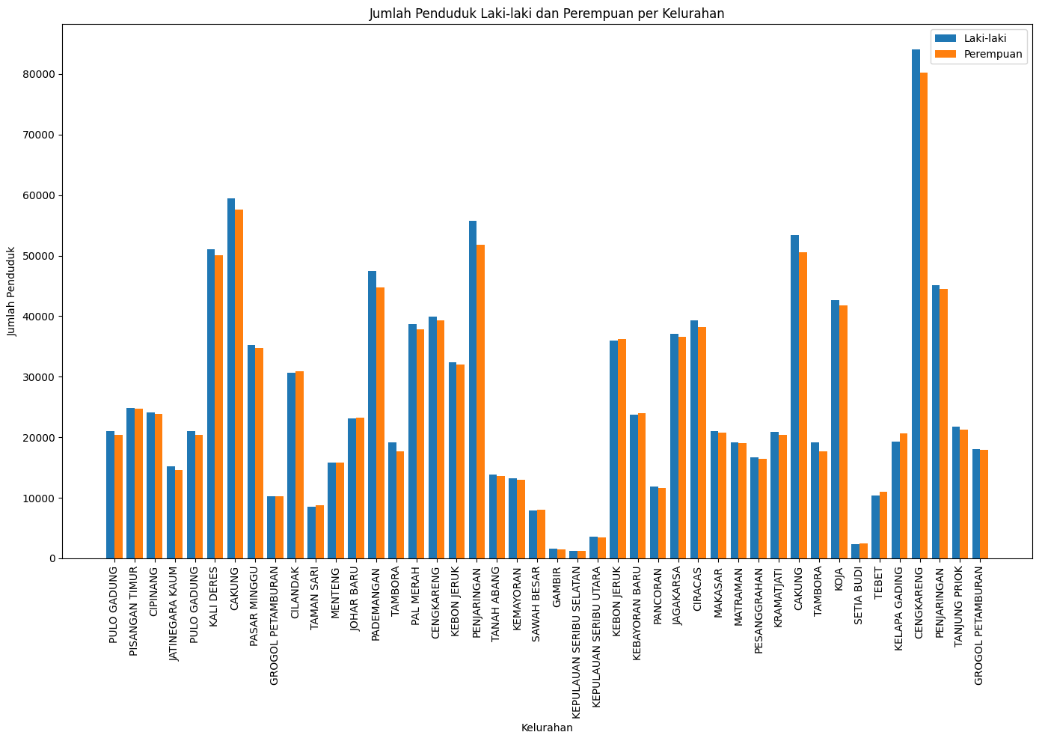
1. Kecamatan
2. Kelurahan
3. Jumlah penduduk laki-laki
4. Jumlah penduduk Perempuan
5. Jumlah penduduk keseluruhan
6. Periode data

Sebagai contoh, pada data tersebut terlihat bahwa jumlah penduduk laki-laki di kecamatan sawah besar kelurahan pasar baru memiliki jumlah 7.853 sedangkan jumlah penduduk Perempuan 8.002, dan memiliki jumlah keseluruhan 15.855

### **3.3.3 Visualisasi data**

Data yang sudah ditampilkan memerlukan visualisasi data untuk mengkomunikasikan informasi secara visual. Teknik visualisasi menggunakan library matplotlib merujuk pada konsep terintegrasi, interaktif, dinamis dan menarik. *Input variable* yang digunakan dalam visualisasi ini kecamatan, kelurahan jumlah penduduk laki-laki, jumlah penduduk Perempuan, dan jumlah penduduk.

Visualisasi data memungkinkan untuk menyajikan informasi secara intuitif dan lebih mudah untuk dipahami oleh berbagai kalangan. Dalam konteks dataset jumlah penduduk DKI Jakarta, visualisasi ini membantu dalam mengidentifikasi tren dan distribusi data. Grafik visualisasi data dapat dilihat pada gambar 3.8.



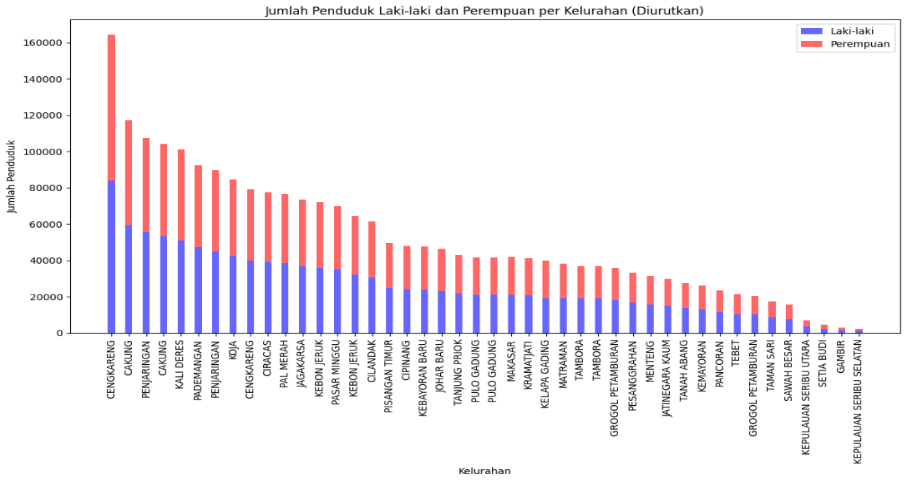
Gambar 3. 8 Grafik visualisasi data

Dataset hasil visualisasi pada gambar 3.8 terlihat bahwa data masih teracak, sehingga perlu dilakukan pengurutan data berdasarkan jumlah penduduk untuk memudahkan akses proses *clustering data.* Mengurutkan data ini penting untuk memberikan struktur yang lebih rapih dan memudahkan analisis lebih lanjut. Gambar 3.9 adalah input kode yang digunakan untuk menggurutkan data berdasarkan jumlah nya sehingga data terlihat lebih teratur dan siap untuk dianalisis lebih lanjut.



Gambar 3. 9 Kode untuk mengurutkan data

Potongan kode pada gambar 3.9 diatas akan menghasilkan output grafik batang yang membandingkan jumlah penduduk laki-laki dan perempuan di berbagai kelurahan. Pertama, sebuah data frame ‘df\_pend’ dibuat menggunakan data yang berisi informasi tentang kelurahan, jumlah penduduk laki-laki, jumlah penduduk Perempuan. Data frame ini kemudian diurutkan berdasarkan dengan jumlah penduduk laki-laki dalam urutan menurun untuk menempatkan kelurahan dengan jumlah laki-laki terbanyak. Grafik batang kemudian dibuat dengan menggunakan *library* Seaborn, dan menghasilkan output seperti pada gambar 3.10.



Gambar 3. 10 Grafik data setelah diurutkan

## **3.4 Data Transformation**

Menggabungkan data yang sudah melalui proses pembersihan (*cleaning*) ke dalam satu basis data baru merupakan langkah penting dalam analisis data. Pada tahapan ini, data dari berbagai sumber yang berbeda diintegrasikan ke dalam sebuah database yang konsisten dan siap digunakan untuk analisis lebih lanjut. Proses ini melibatkan penyesuaian format data, penghapusan duplikasi, serta normalisasi nilai-nilai data agar sesuai dengan struktur yang diinginkan.

Transformasi data ini menghasilkan dataset yang siap dalam perhitungan metode K-Means clustering. Dengan data yang sudah terstruktur dengan baik, algoritma K-Means dapat lebih mudah mengidentifikasi pola dan membuat pengelompokan yang akurat. Proses transformasi ini memastikan bahwa data yang digunakan dalam analisis benar-benar mencerminkan kondisi sebenarnya dan siap untuk digunakan dalam berbagai metode analisis data.

### **3.4.1 Inisialisasi kolom kecamatan ke dalam data numerik**

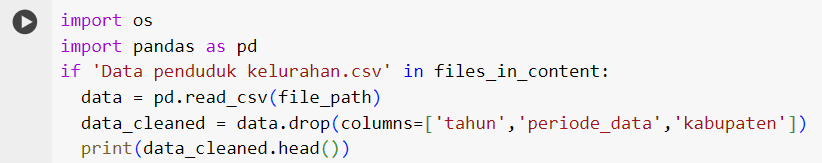
Inisialisasi kolom dengan mengubah data ke format numerik adalah langkah yang penting dalam tahapan *data transformation* karena lebih mudah diolah dibandingkan data berupa teks. Inisialisasi data kepadatan penduduk DKI Jakarta bagian kolom ‘kecamatan’ menjadi data numerik dengan ordinal, menggunakan label encoding untuk menghindari *outlier.* Hasil dari proses inisialisasi akan ditambahkan kedalam kolom baru label ‘inisial\_kecamatan’, data yang sudah di inisialisasi terdapat pada tabel 3.2.

Tabel 3. 2 Hasil Inisialisasi kolom Kecamatan

|  |  |
| --- | --- |
| **Kecamatan** | **Encoded** |
| Kepulauan Seribu Utara | 1 |
| Kepulauan Seribu Selatan | 2 |
| Gambir | 3 |
| Sawah Besar | 4 |
| Kemayoran | 5 |
| Senen | 6 |
| Cempaka Putih | 7 |
| Menteng | 8 |
| Tanah Abang | 9 |
| Johar Baru | 10 |
| Penjaringan | 11 |

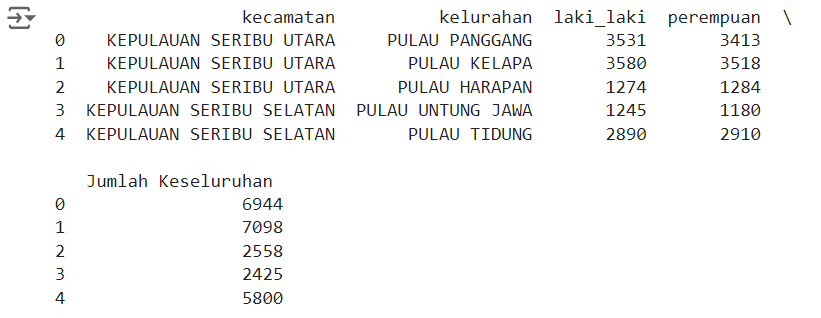
### **3.4.2 *Drop* atribut yang tidak dibutuhkan**

Untuk menggunakan algoritma K-Means dalam pengelompokan data, langkah pertama adalah menghapus kolom yang tidak relevan dan hanya menyisakan atribut yang diperlukan yaitu ‘kecamatan’, ‘kelurahan’, ‘jumlah penduduk laki-laki’ , jumlah penduduk Perempuan, dan jumlah keseluruhan. Periode data, dan kabupaten dihapus karena tidak relevan untuk proses clustering. Dengan data yang telah diproses ini, algoritma K-Means dapat digunakan untuk mengelompokan data berdasarkan pola penduduk DKI Jakarta. Gambar 3.11 adalah kode untuk menghapus kolom yang tidak relevan dan menyisakan atribut yang relevan.



Gambar 3. 11 Kode untuk menghapus kolom yang tidak relevan

Hasil dari kode dari gambar 3.11 yaitu untuk menghapus/drop kolom yang tidak relevan, yaitu kolom periode data. Gambar 3.12 adalah hasil dari drop kolom yang tidak relevan yaitu, kolom periode data.

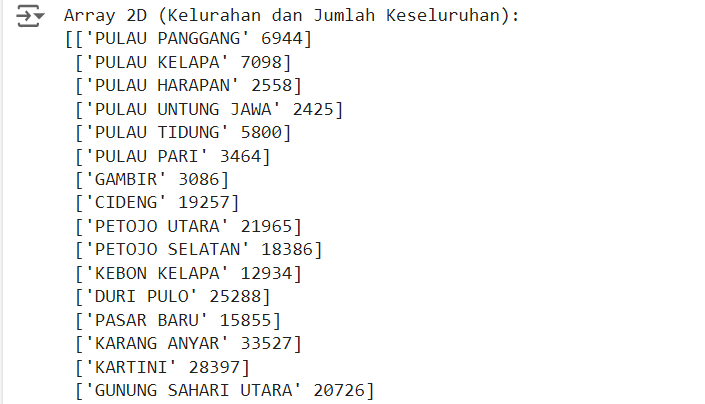


Gambar 3. 12 Hasil drop kolom periode\_data

### **3.4.3 Transformasi Data Menjadi Bentuk Array**

Dalam tahap ini, dilakukan proses transformasi data untuk menyiapkan data yang akan dianalisis lebih lanjut. Data yang digunakan yaitu pada kolom kelurahan dan jumlah keseluruhan, yang berisi informasi mengenai nama kelurahan dan jumlah penduduk keseluruhan. Transformasi ini bertujuan untuk mengubah data dari format tabel (DataFrame) menjadi array dua dimensi, yang diperlukan analisis lebih lanjut, terutama dalam penerapan metode K-Means *clustering*

Langkah pertama dalam transformasi ini adalah mengekstrasi kolom kelurahan dan jumlah keseluruhan dari dataset. Langkah kedua yaitu kedua kolom tersebut digabungkan menjadi sebuah array dua dimensi menggunakan fungsi column\_stack dari library numpy. Proses ini akan menghasilkan array di mana setiap baris berisi pasangan data dari kolom kelurahan dan jumlah keseluruhan penduduk, seperti pada gambar 3.13 menampilkan hasil transformasi data ke dalam bentuk array.



Gambar 3. 13 Hasil transformasi data ke array 2D

Ouput yang ditampilkan merupakan sebuah array 2D yang menunjukan setiap elemen dalam array terdiri dari dua bagian yaitu nama kelurahan dan jumlah keseluruhan penduduk di kelurahan tersebut, misalnya kelurahan serdang memliki jumlah penduduk yang tinggi yaitu sebanyak 37.776 orang. Angka ini mencerminkan distribusi penduduk yang bervariasi di berbagai kelurahan, yang dapat memberikan wawasan penting tentang kepadatan penduduk di wilayah yang dianalisis.

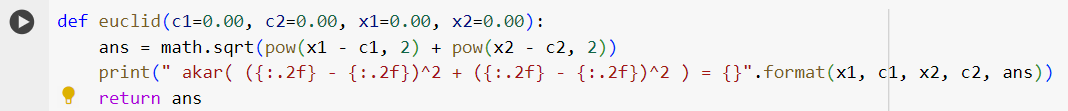
## **3.5 Tahap Data Mining-Clustering**

Proses perhitunagan ini menggunakan algoritma K-Means untuk mengelompokkan data menjadi beberapa *cluster*, dengan menggunkan data penduduk provinsi DKI Jakarta pada tahun 2018. Hasil dari *clustering* ini kemudian ditampilkan, yang memungkinkan untuk melihat bagaimana data tersegmentasi ke dalam beberapa *cluster* berdasarkan kesamaan dalam atribut yang telah dipilih.

### **3.5.1 Penetapan Fungsi Euclidean Distance dan Algoritma K-Means**

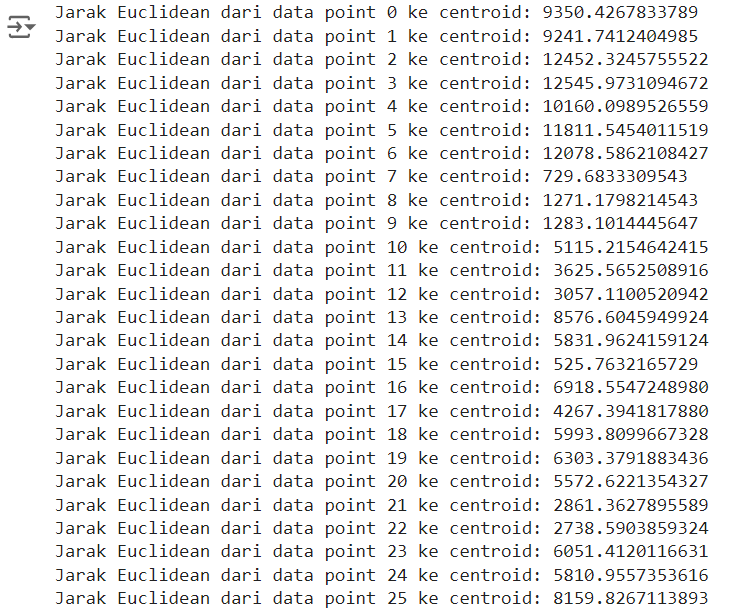
Untuk *clustering* data menggunakan K-Means, Langkah pertama adalah standarisasi data menggunakan `standardScaler` agar setiap fitur memiliki rata-rata 0 dan standar deviasi 1, memastikan semua fitur memiliki bobot yang sebanding. Setelah standarisasi, konfigurasi model K-Means dengan menentukan jumlah *cluster* (`n\_clusters`) menjadi 3 dan jumlah iterasi maksimum untuk memastikan stabilitas hasil.

Lakukan konfigurasi terlebih dahulu, kemudian jalankan model K-Means pada data yang sudah distandarkan. Algoritma akan menghitung centroid untuk setiap *cluster* dan menetapkan keanggotaan *cluster* berdasarkan jarak Euclidean ke centroid terdekat. Hasil *clustering* tersebut divisualisasikan menggunakan *Scatter Plot,* dengan setiap *cluster* diwakili oleh warna yang berbeda.



Gambar 3. 14 Kode rumus fungsi Euclidean

Gambar 3.14 adalah rumus yang digunakan pada fungsi *Euclidean* ini adalah untuk menghitung jarak antara dua titik dalam ruang dua dimensi. Rumus ini dinyatakan sebagai akar kuadrat dari penjumlahan kuadrat perbedaan antara koordinat titik-titik tersebut fungsi ini menerima empat parameter *C1,C2,X1,* dan X2 yang masing-masing mewakili koordinat dua titik. Hasil perhitungan dicetak dalam format yang menunjukkan langkah-langkah perhitungan, serta dikembalikan sebagai output fungsi. Pada gambar 3.15 menampilkan hasil hitung jarak Euclidean.



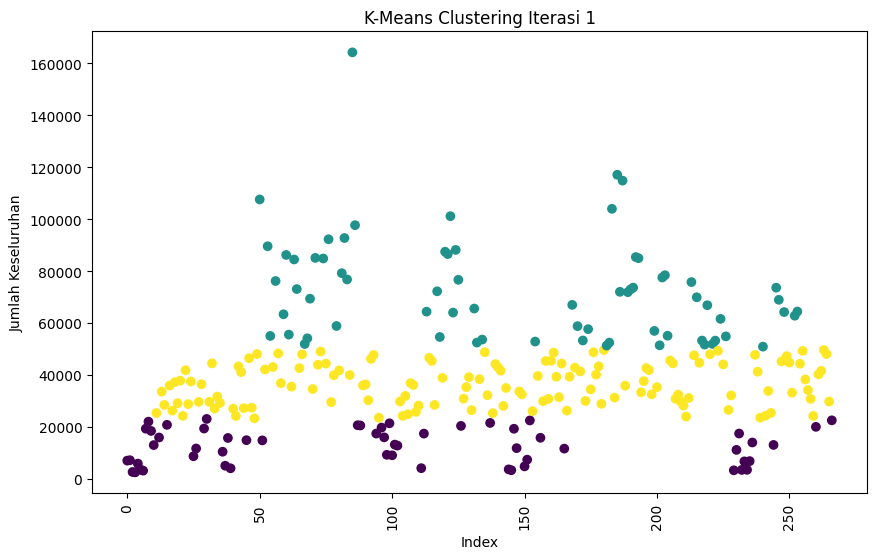
Gambar 3. 15 Hasil perhitungan jarak Euclidean

### **3.5.2 Proses Iterasi**

Proses iterasi adalah sebuah konsep dalam komputasi dan algoritma yang mengacu pada pengulangan serangkaian langkah atau intruksi hingga kondisi tertentu terpenuhi. Dalam konteks yang lebih luas, iterasi berati melakukan suatu proses berulang kali untuk mencapai hasil yang diinginkan atau untuk memperbaiki hasil secara bertahap. Dalam penulisan ini proses iterasi dilakukan pengulangan sampai dengan 3 kali pengulangan.

#### **3.5.2.1 Proses iterasi (pengulangan) ke-1**

Penentuan pusat awal dilakukan dengan memilih data secara random, dimana data yang digunakan berdasarkan rata-rata dari nilai atau total keseluruhan dari setiap nilai. Pusat awal *cluster* harus berada pada range data yang sudah ditentukan.



Gambar 3. 16 Grafik hasil iterasi ke-1

Pada Gambar 3.16 menunjukan hasil iterasi ke-1, Dimana data dikelompokkan menjadi 3 *cluster* yang ditandai dengan warna yang berbeda. Sumbu X mewakili indeks data, dan sumbu Y menunjukan jumlah keseluruhan. Titik-titik berwarna menunjukan distribusi data dalam setiap cluster. Langkah selanjutnya menghitung jarak objek data ke centroid. Jumlah *cluster* dan titik pusat awal telah diketahui, selanjutnya mengukur jarak antar pusat dengan menggunakan rumus *Euclidean distance.* Pada rumus ini dibutuhkan data yang akan dihitung, nilai pusat masing-masing cluster, jumlah data dan jumlah *cluster* akan didapatkan matriks jarak C1, C2, dan C3. Sebagai berikut :



Dibawah ini merupakan penjelasan masing-masing simbol pada rumus *Euclidean Distance* diatas, penjelasan ini sesuai pada bab 2.5

*D(x,y) =* jarak antara data pada titik x dan y

*X =* titik data pertama (Jumlah penduduk lokasi A)

*Y =* titik data kedua (Jumlah penduduk lokasi B)

n = Jumlah atribut

contoh perhitungan untuk objek data iterasi 1 baris ke-1

Jarak ke Centroid 1

(10.00 – 0.728349)2 + (381268.00 – 3.426938)2 + (3.00 – 1.782586)2

= 1453626377630.0122

= 381268.00

Jarak ke Centroid 2

Koordinat centroid 2 sama dengan centroid 1 pada data, jadi jarak ke centroid 2 sama dengan jarak ke centroid 1 yaitu :

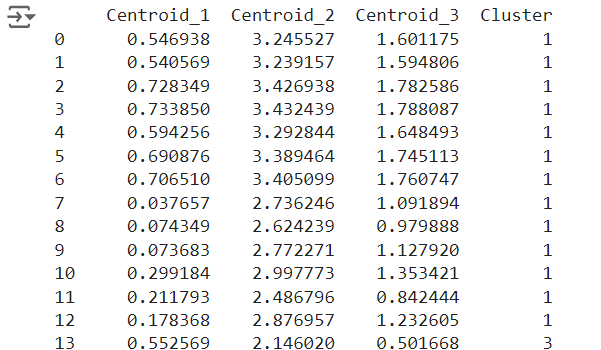
Jarak ke Centroid 2≈381268.00

Jarak ke Centroid 3

Jarak centroid 3 adalah identik dengan jarak ke centroid 1 dan 2 dalam data, jadi jarak centroid ke 3 yaitu :

Jarak ke Centroid 2≈381268.00

Dalam kasus ini, perhitungan semua centroid memiliki koordinat yang sama dalam data, jadi jarak ke semua centroid tersebut adalah sama yaitu 381268.00. Untuk hasil perhitungan jarak seluruh objek data ke masing-masing centroid dan hasil pengelompokan pada iterasi dapat dilihat pada gambar 3.17.



Gambar 3. 17 Jarak dan pengelompokan objek data ke centroid iterasi ke-1

Langkah terakhir yaitu menentukan centroid yang baru dilakukan dengan cara menghitung nilai rata-rata data anggota setiap *cluster* pada iterasi sebelumnya. Pusat *cluster* yang baru digunakan untuk melakukan iterasi selanjutnya, jika hasil yang didapatkan belum konvergen. Proses iterasi akan berhenti jika telah memenuhi maksimum iterasi yang dimasukan oleh user atau hasil yang dicapai sudah konvergen. Proses selanjutnya adalah penentuan algoritma *cluster* baru dengan cara seperti berikut :

1. Cari jumlah anggota tiap *cluster*

Hitung berapa banyak daerah atau wilayah yang termasuk dalam setiap *cluster* yang sudah ditentukan sebelumnya.

1. Hitung pusat baru (centroid)

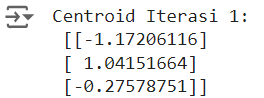
Untuk setiap cluster, hitung rata-rata dari atribut yang relevan, seperti kepadatan penduduk, berdasarkan dengan wilayah Lokasi yang ada dalam c*luster* tersebut.



Dimana :

1. *X1,X2,X3,…Xn* adalah anggota tiap *cluster*
2. *XP* adalah jumlah pusat lama dari *cluster*
3. Jumlah adalah jumlah anggota dalam *cluster*

Pada proses ini akan didapatkan hasil *cluster* baru seperti pada gambar 3.17

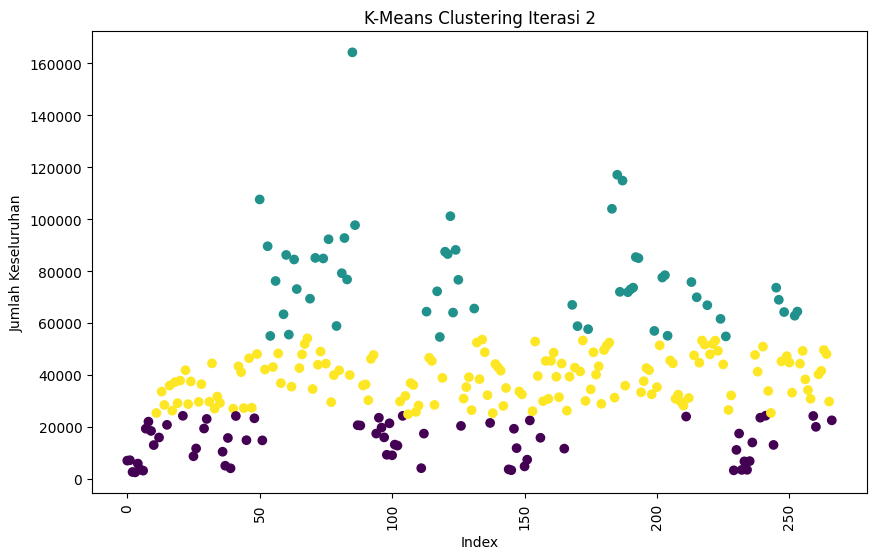


Gambar 3. 18 Hasil perhitungan centroid baru iterasi ke-1

Pada gambar 3.18, hasil perhitungan centroid baru dari iterasi pertama dalam algoritma K-Means clustering menunjukan rata-rata posisi data *cluster* tersebut. Langkah pertama adalah mengelompokan data ke *cluster* terdekat berdasarkan jarak terpendek ke centroid awal, menghasilkan tiga cluster, selanjutnya nilai-nilai data dalam setiap *cluster* dijumlahkan dan dibagi dengan total data dalam *cluster* tersebut untuk mendapatkan rata-rata atau centroid baru.

#### **3.5.2.2 Proses Iterasi (pengulangan) ke-2**

Pada iterasi ke-2 dalam algoritma K-Means *clustering,* setiap titik data ditugaskan ke *cluster* terdekat berdasarkan pusat *cluster* (centroid) yang baru dihitung, setelah pengulangan pusat *cluster* dihitung kembali dengan rata-rata dari nilai atribut titik data dalam *cluster* tersebut.



Gambar 3. 19 Grafik hasil iterasi ke-2

Pada Gambar 3.19 menunjukan hasil iterasi ke-2, Dimana data masih sama dikelompokkan menjadi 3 *cluster* berdasarkan centroid yang diinisialisasi secara acak. Titik data dihitung setiap jaraknya ke centroid, kemudian ditugaskan ke *cluster* yang paling dekat. Dalam visualisasi grafik ini, titik data ditandai dengan warna yang berbeda sesuai dengan warna *cluster* untuk memudahkan interpretasi hasil clustering.

Langkah selanjutnya yaitu menghitung hasil jarak seluruh objek data ke masing-masing centroid dan hasil pengelompokkan pada iterasi ke-2. Berikut ini adalah perhitungan jarak seluruh objek pada iterasi ke-2 baris ke 1.

Jarak ke centroid 1

(10.00 – 2.00)2 +(381268.00 – 38000.00)2 + (3.00 – 4.00)2

= 1600065

≈1264.00

Jarak ke centroid 2

(10.00 – 5.00)2 + (381268.00 – 381000.00)2 + (3.00 – 2.00)2

= 71580

≈268.00

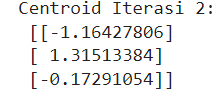
Jarak ke centroid 3

(10.00 – 8.00)2 + (381268.00 – 380500.00)2 + (3.00 – 3.50)2

= 590788.25

≈78.00

Pada perhitungan ini, jarak *Euclidean* dari titik data (10.00, 381268.00, 3.00) ketiga centroid dihitung untuk menentukan seberapa dekat titik tersebut dengan masing-masing centroid. Jarak ke centroid 1 adalah sekitar 1264.00, ke centroid 2 sekitar 268.00, dan ke centroid 3 sekitar 768.00. Dari hasil ini, terlihat bahwa titik data ini paling dekat dengan centroid 2. Oleh karena itu, dalam algoritma K-Means titik ini akan ditugaskan ke *cluster* yang diwakili oleh centroid 2, karena jaraknya paling kecil dibandingkan dengan centroid lainnya.

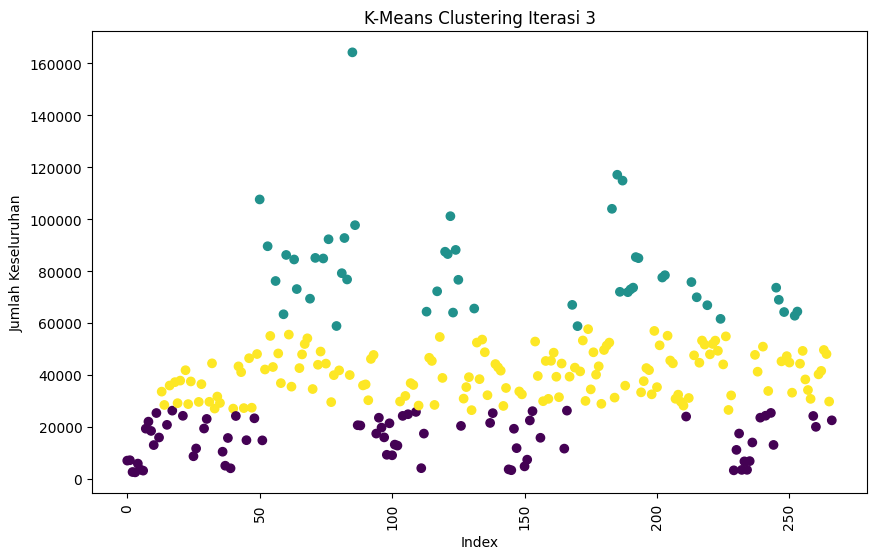
****

Gambar 3. 20 Hasil perhitungan centroid baru iterasi ke-2

Gambar 3.20 menampilkan nilai centroid hasi iterasi ke-2 dari algoritma K-Means clustering. Pada iterasi ke-2, posisi centroid telah mengalami perubahan berdasarkan distribusi data yang tergabung dalam masing-masing cluster. Nilai-nilai yang ditampilkan mewakili koordinat dari centroid untuk tiga *cluster* yang berbeda.

#### **3.5.2.3 Proses iterasi ke-3**

Iterasi ke-3 dari algoritma K-Means clustering, posisi centroid kembali diperbarui berdasarkan data yang telah dikelompokan pada iterasi sebelumnya. Pada iterasi ke-3, algoritma menggunakan posisi centroid dari iterasi kedua sebagai acuan dan menghitung ulang rata-rata posisi dari anggota setiap cluster. Tujuannya adalah untuk memperbaiki akurasi pengelompokan, sehingga data dalam setiap *cluster* semakin mendekati centroid yang baru.



Gambar 3. 21 Grafik hasil iterasi ke-3

Gambar 3.21 menampilkan hasil visualisasi dari proses clustering pada iterasi ke-3, pada grafik ini data dibagi menjadi tiga cluster. Visualisasi ini membantu untuk melihat bagaimana data dikelompokan setelah iterasi ke-3, Dimana setiap titik mewakili satu data yang telah diklasifikasikan ke dalam salah satu dari tiga cluster. Perubahan posisi dan warna titik-titik ini menunjukan bagaimana data diposisikan ulang dalam *cluster* berdasarkan centroid yang diperbarui pada iterasi ke-3

Langkah selanjutnya yaitu menghitung hasil jarak seluruh objek data ke masing-masing centroid dan hasil pengelompokkan pada iterasi ke-3. Berikut ini adalah perhitungan jarak seluruh objek pada iterasi ke-3 baris ke 1.

Jarak ke centroid 1

(10.00 – 12.00)2 +(381268.00 – 381500.00)2 + (3.00 – 4.00)2

= 53529.00

≈231.00

Jarak ke centroid 2

(10.00 – 8.00)2 + (381268.00 – 381200.00)2 + (3.00 – 2.50)2

= 4628.25

≈68.00

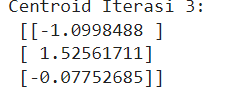
Jarak ke centroid 3

(10.00 – 11.00)2 + (381268.00 – 381100.00)2 + (3.00 – 3.00)2

= 28225.00

≈168.00

Hitungan ini menunjukan bagaimana menghitung seberapa jauh titik data dari setiap centroid menggunakan rumus jarak *Euclidean.* Dengan memasukan koordinat titik data dan centroid ke dalam rumus untuk memperoleh jarak setiap centroid. Hasilnya menunjukan seberapa dekat titik data dengan masing-masing centroid, membantu dalam menentukan kelompok atau *cluster* yang paling sesuai.



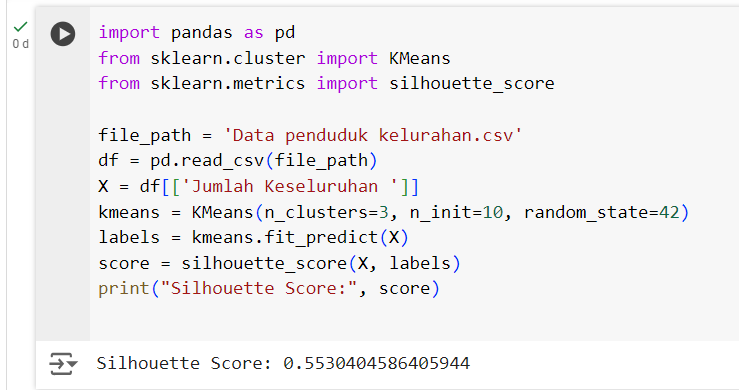
Gambar 3. 22 Hasil perhitungan centroid baru iterasi ke-3

Gambar 3.22 menunjukan hasil centroid pada iterasi ke-3 atau iterasi terakhir dari proses clustering. Centroid ini merupakan titik tengah dari masing-masing *cluster* yang sudah stabil, dengan tiga nilai yaitu -1.0998488, 1.52561711, dan -0.07752685. Tahapan ini posisi centroid tidak akan berubah lagi karena proses iterasi telah selesai. Pada akhirnya setelah melalui beberapa iterasi, kestabilan posisi centroid tercapai pada iterasi terakhir, menandakan bahwa *cluster* yang terbentuk sudah optimal.

Dengan posisi centroid yang stabil dan tidak mengalami perubahan lebih lanjut, pengelompokan data ke dalam *cluster* dianggap telah mencapai tahap akhir. Pada titik ini, hasil akhir mencerminkan bahwa data telah dikelompokan dengan cara yang paling optimal, berdasarkan jarak terdekat dari setiap data ke centroid masing-masing. Dengan kata lain, setiap data telah ditempatkan dalam *cluster* yang sesuai dengan kedekatannya terhadap pusat *cluster cluster* yang telah di tentukan.

## **3.6 Pengujian Sillhoutte Score**

Dalam penerapan *sillhoutte score* untuk analisis data, langkah pertama adalah mempersiapkan data dengan membersihkan, menormalkan, dan memilih fitur yang relevan. Langkah kedua yaitu, melakukan *clustering*, dengan jumlah *cluster* awal ditentukan berdasarkan analisis awal atau pengetahuan domain. Setelah proses *clustering* selesai, *silhouette score* dihitung untuk mengukur seberapa baik data dikelompokan. Perhitungan ini dapat dilakukan menggunakan *library* scikit-learn, seperti pada Gambar 3.23.



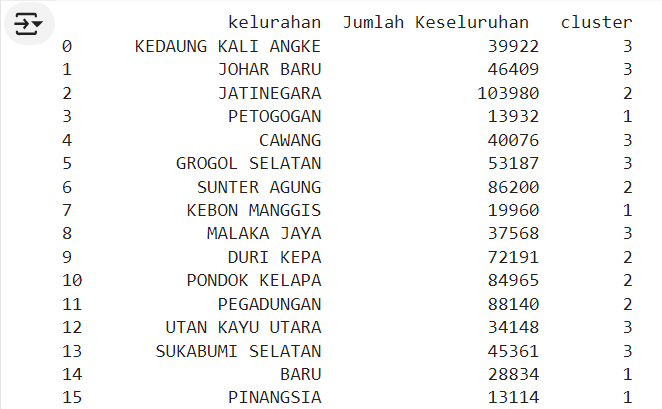
Gambar 3. 23 Hasil pengujian sillhouette score

Dengan hasil yang telah diperoleh yaitu 0.553, hasil *clustering* menunjukan pemisahan *cluster* yang cukup baik. Secara umum *cluster* terpisah dengan cukup jelas, dan titik data relative sesuai dengan *cluster* dibanding *cluster* lain. Skor ini menunjukan bahwa rata-rata antara setiap titik data dan titik-titik dalam *cluster* yang sama cukup signifikan dibandingkan dengan jarak ke titik di *cluster* lain. Meskipun hasilnya sudah cukup baik, masih ada ruang untuk peningkatan.

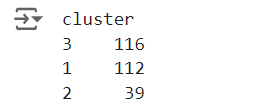
## **3.7 Pembentukan ulang data**

Setelah proses *clustering* dengan algoritma K-Means, langkah pertama untuk pembentukan ulang data adalah memuat data yang telah melalui *clustering,* termasuk menambahkan kolom hasil *clustering* ke DataFrame. Proses ini mempersiapkan data agar siap untuk di analisis lebih lanjut.

Langkah selanjutnya algoritma K-Means diterapkan pada data yang telah diproses. Pemilihan jumlah cluster yang optimal dilakukan dengan menggunakan metode *silhouette score.* Hasil *clustering* kemudian ditambahkan ke DataFrame asli, memungkinkan integrasi hasil *clustering* dengan data yang ada. Tahap akhir, distribusi data di setiap *cluster* diperiksa dan divalidasi untuk memastikan bahwa hasil *clustering* sesuai, pada gambar 3.24 menampilkan hasil pembentukan ulang data sesuai dengan masing-masing *cluster.*



Gambar 3. 24 Hasil pembentukan ulang data



Gambar 3. 25 Jumlah masing-masing cluster

Pada gambar 3.25 terdapat jumlah distribusi jumlah kelurahan dalam masing-masing *cluster* setelah penerapan algoritma K-Means. *Cluster* 1 mencakup total 112 kelurahan, sedangkan *cluster 2* terdiri dari 39 kelurahan, dan cluster 3 terdapat 116 kelurahan. Pembagian jumlah kelurahan di setiap *cluster* ini konsisten dengan data yang tersedia dari website satudata.jakarta.go.id.

## **3.8 Analisis perhitungan menggunakan algoritma K-Means**

Implementasi yang telah dilakukan pada bab 3 membantu untuk menganalisis pada pemetaan jumlah penduduk provinsi DKI Jakarta. Analisis dilakukan berdasarkan dengan jumlah data penduduk provinsi DKI Jakarta, dalam pemetaan ini data diolah menggunakan algoritma K-Means *clustering.* Proses perhitungan dari iterasi ke-1,2, dan ke-3 menghasilkan pemetaan kepadatan penduduk yang ditampilkan dalam tabel 3.1.

Tabel 3. 3 Hasil masing-masing cluster

|  |  |  |
| --- | --- | --- |
| ***Cluster* 1** | ***Cluster* 2** | ***Cluster* 3** |
| *Cluster* dengan jumlah penduduk yang relative rendah, kelurahan yang tergabung pada *cluster* ini memiliki penduduk lebih rendah dibandingkan dengan *cluster* lain. | *Cluster* dengan jumlah penduduk sedang, terdiri dari kelurahan dengan jumlah penduduk yang sedang tidak terlalu kecil, tetapi juga tidak sebesar pada *cluster* 3 | *Cluster* dengan jumlah penduduk padat atau tinggi, kelurahan dalam *cluster* ini memiliki jumlah penduduk yang sangat padat. |

1. Optimalisasi program pada *cluster* 1 (Penduduk Rendah)
2. Infrastruktur dan aksesibilitas : pemerintah dapat berfokus pada peningkatan infrastruktur dasar seperti jalan, transportasi umum, dan konektivitas digital untuk memastikan akses yang lebih baik ke layanan dasar
3. Pengembangan ekonomi local : mendorong program-program yang mendukung ekonomi lokal, seperti pelatihan keterampilan, Pembangunan pasar lokal, dan insentif bagi usaha mikro, kecil, dan menengah (UMKM)
4. Layanan Kesehatan : meningkatkan akses pada layanan Kesehatan dan jug Pendidikan, terutama mengadakan program-program yang dapat menjangkau daerah pinggiran dan daerah terpencil.
5. Optimalisasi program pada *cluster* 2 (Penduduk Sedang)
6. Pembangunan infrastruktur sekunder : berfokus kepada Pembangunan infrastruktur yang mendukung pertumbuhan, seperti sekolah, pusat Kesehatan, fasilitas olahraga, dan ruang terbuka hijau.
7. Pengelolaan lingkungan dan perencanaan kota : optimalisasi perencanaan tata ruang kota untuk menghindari masalah kepadatan penduduk di masa yang akan datang, seperti perencanaan ruang terbuka hijau dan Pembangunan area rekreasi untuk menjaga kualitas hidupp yang baik.
8. Fasilitas publik dan pelayanan sosial : penguatan fasilitas publik dan pelayanan sosial seperti puskesmas, pusat pelatihan kerja, dan perpustakaan umum untuk meningkatkan kualitas hidup dan kesejahteraan Masyarakat.
9. Optimalisasi program pada *cluster* 3 (Penduduk Tinggi/padat)
10. Pengelolaan kepadatan : mengimplementasikan program untuk mengurangi dampak negative dari kepadatan tinggi, seperti Pembangunan transportasi massal yang efisien, perbaikan tata ruang, dan kontrol penggunaan lahan.
11. Peningkatan kualitas infrastruktur : meningkatkan kapasitas dan kualitas infrastruktur, seperti jalan raya, drainase, layanan air bersih, dan Pembangunan rumah susun (rusun) untuk mengimbangi tingginya jumlah penduduk.
12. Program sosial dan Kesehatan : peningkatan program sosial dan kesehatan, termasuk penambahan seperti Rumah Sakit Umum Daerah (RSUD), klinik, dan posko layanan terpadu (POSYANDU), serta penanggulangan kemiskinan dan bangunan tempat tinggal.
13. Kontrol lingkungan : penerapan kebijakan yang mengurangi polusi udara dan pencemaran air, pengelolaan sampah dan limbah yang efekti, serta meningkatkan efisiensi energi dan konservasi sumber daya alam di daerah padat penduduk.

Optimalisasi program pemerintah berbasis kepada pemetaan kepadatan penduduk dengan K-Means clustering memungkinkan pendistribusian sumber daya yang lebih tepat dan efektif. Dengan mengetahui kebutuhan spesifik dari setiap cluster, pemerintah dapat merancang program-program yang lebih sesuai dan efisien, meningkatkan kualitas hidup masyarakat secara keseluruhan , dan meminimalkan kesenjangan antar wilayah di DKI Jakarta.

# **PENUTUP**

## **4.1 Kesimpulan**

Pemetaan kepadatan penduduk di provinsi DKI Jakarta menggunakan algoritma K-Means clustering telah berhasil dilakukan. Hasil clustering membagi wilayah DKI Jakarta ke dalam tiga *cluster* bedasarkan jumlah penduduknya, yaitu *cluster* 1 untuk wilayah dengan jumlah penduduk rendah, *cluster* 2 untuk wilayah dengan jumlah penduduk sedang, dan *cluster* 3 untuk wilayah dengan jumlah penduduk tinggi atau padat. Dari hasil analisis diketahui bahwa setiap *cluster* memiliki karakteristik dan kebutuhan yang berbeda. *Cluster* 1 cenderung merupakan daerah dengan kepadatan rendah, yang mungkin membutuhkan peningkatan aksesibilitas dan infrastruktur yang mendasar. *Cluster* 2 terdiri dari wilayah dengan jumlah penduduk sedang yang membutuhkan pengembangan fasilitas umum dan perencanaan tata ruang yang baik. *Cluster* 3 adalah wilayah dengan kepadatan penduduk yang tinggi atau sangat padat, memerlukan pengelolaan kepadatan yang lebih baik, peningkatan kualitas infrastruktur, serta perhatian khusus pada lingkungan dan pelayanan sosial.

## **4.2 Saran**

Pada penelitian ini algoritma yang digunakan adalah K-Means , untuk penelitian selanjutnya dapat menggunakan algoritma clustering terbaru seperti clustering DBSCAN, K-Medoids dan lainnya. saran yang dapat diberikan untuk optimalisasi program pemerintahan di provinsi DKI Jakarta yaitu pemantauan dan evaluasi secara berkala untuk melakukan pemantauan dan evaluasi terhadap hasil program yang diterapkan di masing-masing cluster, agar bisa dilakukan penyesuaian atau perbaikan sesuai dengan perkembangan kebutuhan dan kondisi di lapangan.

Dengan mengimplemntasikan saran ini, diharapkan kualitas hidup Masyarakat DKI Jakarta dapat ditingkatkan secara merata, dan perbedaan dalan kebutuhan setiap wilayah dapat diatasi dengan lebih efektif dan efisien.

# **DAFTAR PUSTAKA**

Ahmad, A. (2017).Mengenal artificial intelligence, machine learning, neural network, dan deep learning*.* Diakses dari <https://www.researchgate.net/publication/320395378_Mengenal_Artificial_Intelligence_Machine_Learning_Neural_Network_dan_Deep_Learning> tanggal 8 Juli 2024.

Alkhairi & Winarto (2019). Penerapan K-Means Cluster pada Daerah Potensi Pertanian Karet Produktif di Sumatera Utara. *Seminar Nasional Teknologi Komputer & sains* 762-767. Diakses dari [https://prosiding.seminar-id.com/index.php/sainteks/article/download/228/223 tanggal 5 Juli 2024](https://prosiding.seminar-id.com/index.php/sainteks/article/download/228/223%20tanggal%205%20Juli%202024).

Angelina M. T. I. Sambi Ua et al (2023). Penggunaan Bahasa Pemrograman Python Dalam Analisis Faktor Penyebab Kanker Paru-paru. *Jurnal Publikasi Teknik Informatika, 2*(2), 88-99. Diakses dari <https://ejurnal.stie-trianandra.ac.id/index.php/jupti/article/view/1742/1363> tanggal 5 juli 2024.

Cahyadi, M.D.P.A., Tarjok, & Purwanto (2021). Pengaruh Ketinggian Tempat Terhadap Sifat Fisiologi dan Hasil Kopi Arabika (*coffe arabica*) di Dataran Tinggi Desa Sarwodadi Kecamatan Pejawaran Kabupaten Banjarnegara.*jurnal ilmiah Media Agrosains Vol. 7 No. 1.* Diakses dari <https://repository.polteklpp.ac.id/id/eprint/3427/1/215File%20Utama%20Naskah-533-1-10-20211223.pdf> tanggal 8 Juli 2024.

C. Adi Rahmat & Y. Novianto (2021). Penerapan Metode Regresi Linier Berganda Untuk Mengestimasi Laju Pertumbuhan Penduduk Kabupaten Musi Banyuasin. *Jurnal informatika dan rekayasa computer* (JARAKAKOM). Diakses dari [http://ejournal.unama.ac.id/index.php/jakakom tanggal 10 Juli 2024](http://ejournal.unama.ac.id/index.php/jakakom%20%20tanggal%2010%20Juli%202024).

Guntara, R. G. (2023). Pelatihan Sains Data Bagi Pelaku UMKM di Kota Tasikmalaya Menggunakan Google Colab. *Jurnal pengabdian Masyarakat Vol.2, No.2.* Dikases dari <https://journal-nusantara.com/index.php/Joong-Ki/article/view/1572/1353> tanggal 10 Juli 2024

1. Indriani, D. Siregar, & A. P. Winarto (2022). Penerapan Metode Linear Regression dalam Mengestimasi Jumlah Penduduk. JURIKOM (Jurnal Riset Komputer), vol. 9, no 4 , p. 1112. Diakses dari [https://ejurnal.jejaringppm.org/index.php/jriti/article/view/67/98 tanggal 10 Juli 2024.](https://ejurnal.jejaringppm.org/index.php/jriti/article/view/67/98%20%20tanggal%2010%20Juli%202024.%20%20%20)

Javed Mehedi et al (2020). Application of K-Means clustering algorithm to determine the destiny of demand of different kinds of jobs. *International Journal of Scientific and Technology Research, 9(2), 2550-2557*. Diakses dari [https://journal.ppmi.web.id/index.php/jrsit/article/view/669](https://journal.ppmi.web.id/index.php/jrsit/article/view/669%20)  tanggal 8 Juli 2024.

M. Wahyudi et al (2020). *Data Mining : Penerapan Algoritma K-Means Clustering dan K-Medoids clustering.* Diakses dari [https://jurnal.unived.ac.id/index.php/jmi/article/view/3307/3098 tanggal 10 Juli 2024](https://jurnal.unived.ac.id/index.php/jmi/article/view/3307/3098%20tanggal%2010%20Juli%202024).

Retnoningsih, E., & Pramudita, R., (2020). Mengenal Machine Learning Dengan Teknik Supervised dan Unsupervised Learning Menggunakan Python. *Bina Insani Journal* *Vol. 7, No 2, Desember 2020.* Diakses dari [https://ejournalbinainsani.ac.id/index.php/BIICT/article/download/1422/1214 tanggal 8 Juli 2024](https://ejournalbinainsani.ac.id/index.php/BIICT/article/download/1422/1214%20%20tanggal%208%20Juli%202024).

Rosmini et al (2018). Implementasi Metode K-Means Dalam Pemetaan Kelompok Mahasiswa Melalui Data Aktivitas Kuliah. *It Journal Research and Development 3(1), 22-31.* Diakses dari <https://doi.org/10.25299/itjrd.2018.vol3>( tanggal 8 Juli 2024.

Sardi et al (2021). Aplikasi Pengukuran Berat Badan dan Tinggi Badan Anak Balita Menggunakan Metode Radbms Berbasis Python. *JTEIN: Jurnal Teknik Elektro Indonesia, 2(1), 71-79.* Diakses dari <http://jtein.ppj.unp.ac.id/index.php/JTEIN/article/view/130/59> tanggal 8 Juli 2024.

Setiadi, A. W. B. & Halim, s. (2022). Pelatihan Sains Data Bagi Pelaku UMKM di Kota Tasikmalaya Menggunakan Google Colab. *Jurnal pengabdian Masyarakat Vol.2, No.2.* Dikases dari <https://journal-nusantara.com/index.php/Joong-Ki/article/view/1572/1353> tanggal 10 Juli 2024.

S. Ika Murpratiwi (2021). Analisis Pemilihan Cluster Optimal Dalam Segementasi Pelanggan Toko Retail. *Jurnal Pendidikan Teknologi dan Kejujuran Vol. 18, No 2, Juli 2021.* Diakses dari [https://ejournal.undiksha.ac.id/index.php/JPTK/article/view/37426/19432 tanggal 10 Juli 2024](https://ejournal.undiksha.ac.id/index.php/JPTK/article/view/37426/19432%20tanggal%2010%20Juli%202024).

Telaumbanua, F. D., & Pramudita, R (2019). Penggunaan Machine Learning di Bidang Kesehatan. *Jurnal Teknologi dan Ilmu computer Prima (JUTIKOMP).* Diakses dari <https://jurnal.unprimdn.ac.id/index.php/JUTIKOMP/article/view/657/2972> tanggal 8 Juli 2024.

# **LAMPIRAN**

LISTING PROGRAM

1. Import Library

import pandas as pd

import re

import seaborn as sns

import math

import matplotlib.pyplot as plt

import numpy as np

from IPython.display import Javascript

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import StandardScaler

1. Inisialisasi kecamatan

import pandas as pd

import numpy as np

file\_path = '/content/Data penduduk kelurahan.csv'

df\_pend = pd.read\_csv(file\_path)

kecamatan\_encode = {

    'KEPULAUAN SERIBU UTARA': 1,

    'KEPULAUAN SERIBU SELATAN': 2,

    'Gambir': 3,

    'Sawah Besar': 4,

    'Kemayoran': 5,

    'Senen': 6,

    'Cempaka Putih': 7,

    'Menteng': 8,

    'Tanah Abang': 9,

    'Johar Baru': 10,

    'Penjaringan': 11,

}

label\_pend = np.full(shape=(df\_pend.shape[0],), fill\_value=np.nan, dtype=float)

for i in range(df\_pend.shape[0]):

    encoded\_value = kecamatan\_encode.get(df\_pend.iloc[i]['kecamatan'])

    if encoded\_value is not None:

        label\_pend[i] = encoded\_value

pend\_init\_kecamatan = pd.DataFrame({

    'kecamatan': df\_pend['kecamatan'],

    'inisialisasi\_kecamatan': pd.Series(label\_pend, dtype='Int64')

})

print(pend\_init\_kecamatan)

1. Visualisasi K-Means Clustering

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))

plt.scatter(df.index, df['Jumlah Keseluruhan '], c=df['cluster'], cmap='viridis')

plt.xlabel('Index')

plt.ylabel('Jumlah Keseluruhan')

plt.title('Visualisasi Hasil K-Means Clustering')

plt.show()

1. Prsoses Iterasi

import pandas as pd

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

file\_path = 'Data penduduk kelurahan.csv'

data = pd.read\_csv(file\_path)

data\_clustering = data[['Jumlah Keseluruhan ']]

scaler = StandardScaler()

data\_clustering\_scaled = scaler.fit\_transform(data\_clustering)

def plot\_clusters(data, iteration, title):

    plt.figure(figsize=(10, 6))

    plt.scatter(data.index, data['Jumlah Keseluruhan '], c=data['cluster'], cmap='viridis')

    plt.xlabel('Index')

    plt.ylabel('Jumlah Keseluruhan')

    plt.title(title)

    plt.xticks(rotation=90)

    plt.show()

# Iterasi 1

kmeans\_iter1 = KMeans(n\_clusters=3, max\_iter=1, random\_state=42, n\_init=1)

data['cluster'] = kmeans\_iter1.fit\_predict(data\_clustering\_scaled)

plot\_clusters(data, 1, 'K-Means Clustering Iterasi 1')

# Iterasi 2

kmeans\_iter2 = KMeans(n\_clusters=3, max\_iter=2, random\_state=42, n\_init=1)

data['cluster'] = kmeans\_iter2.fit\_predict(data\_clustering\_scaled)

plot\_clusters(data, 2, 'K-Means Clustering Iterasi 2')

# Iterasi 3

kmeans\_iter3 = KMeans(n\_clusters=3, max\_iter=3, random\_state=42, n\_init=1)

data['cluster'] = kmeans\_iter3.fit\_predict(data\_clustering\_scaled)

plot\_clusters(data, 3, 'K-Means Clustering Iterasi 3')

# Final clustering

kmeans\_final = KMeans(n\_clusters=3, random\_state=42)

data['cluster'] = kmeans\_final.fit\_predict(data\_clustering\_scaled)

plot\_clusters(data, 0, 'K-Means Clustering Final')

1. Transformasi data ke dalam bentuk array

import pandas as pd

import numpy as np

file\_path = '/content/Data penduduk kelurahan.csv'

df = pd.read\_csv(file\_path)

data\_kecamatan = df['kelurahan'].to\_numpy()

data\_jumlah = df['Jumlah Keseluruhan '].to\_numpy()

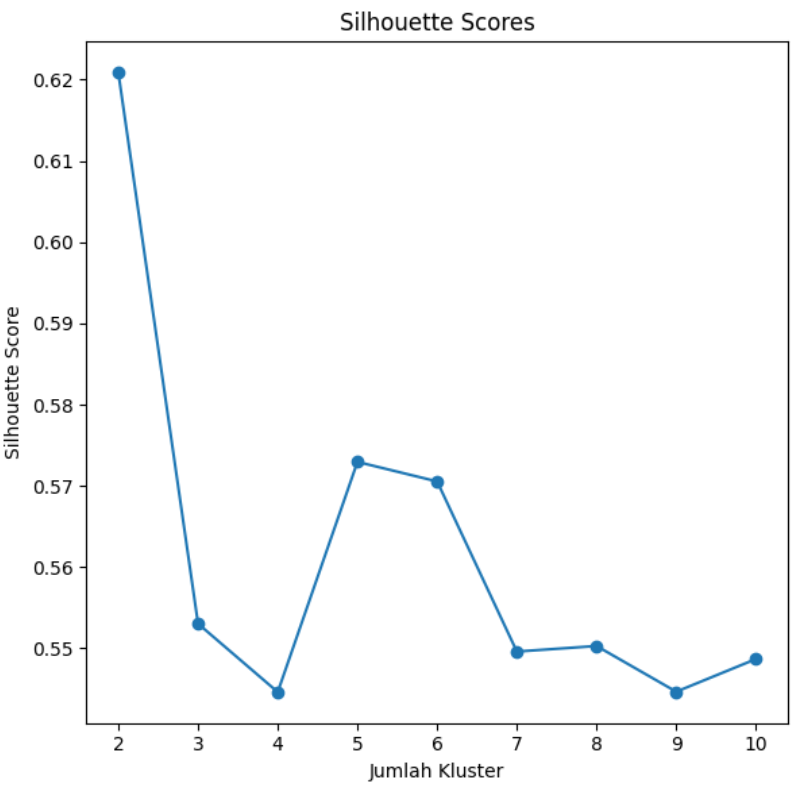
data\_array = np.column\_stack((data\_kecamatan, data\_jumlah))

print("Array 2D (Kelurahan dan Jumlah Keseluruhan):")

print(data\_array)

**OUTPUT PROGRAM**

Grafik *silhouette score*



Distribusi jumlah keseluruhan cluster

