

LAPORAN TUGAS AKHIR

# Komputasi Matematika

*Analisis Clustering Menggunakan Algoritma K-Means  
pada Dataset Heart Failure Clinical Record*



**Kelompok 3 - 2023F**

Ainur Rahman Hidayat	- 23030214033
Maeva Wulandari	- 23030214047
Ganis Amalia Feronika	- 23030214082
Nabilah Nesya Adiarni Azzahra	- 23030214098
Pricila Sisi Austin Soko	- 23030214119

Dosen Pengampu:

Dimas Avian MAULANA, M.Si.

KEMENTERIAN PENDIDIKAN TINGGI, SAINS, DAN TEKNOLOGI  
UNIVERSITAS NEGERI SURABAYA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
PROGRAM STUDI S1 MATEMATIKA

9 Juli 2025

# Daftar Isi

<b>1</b>	<b>Pendahuluan</b>	<b>3</b>
1.1	Latar Belakang . . . . .	3
1.2	Rumusan Masalah . . . . .	4
1.3	Tujuan . . . . .	5
1.4	Manfaat . . . . .	5
<b>2</b>	<b>Tinjauan Pustaka</b>	<b>7</b>
2.1	Gagal Jantung (Heart Failure) . . . . .	7
2.2	Dataset <i>Heart Failure Clinical Record</i> . . . . .	8
2.3	Metode <i>Clustering</i> . . . . .	9
2.4	Algoritma K-Means dalam Analisis Data Medis . . . . .	10
<b>3</b>	<b>Metode Penelitian</b>	<b>11</b>
3.1	Deskripsi Penelitian dan Dataset . . . . .	11
3.2	Variabel Penelitian . . . . .	11
3.3	Instrumen Penelitian . . . . .	12
3.4	Teknik Pengumpulan Data . . . . .	15
3.5	Teknik Analisis Data . . . . .	16
3.6	Algoritma Penelitian . . . . .	17
<b>4</b>	<b>Hasil dan Pembahasan</b>	<b>18</b>
4.1	<i>Pre-Processing</i> Data . . . . .	18
4.1.1	Pengecekan <i>Missing Value</i> . . . . .	18
4.1.2	Reduksi Data . . . . .	19
4.1.3	Standarisasi Data . . . . .	21
4.2	Penerapan K-Means Clustering . . . . .	23
4.2.1	Penerapan Algoritma Menggunakan Cara Manual . . . . .	23
4.2.2	Penerapan Algoritma Menggunakan Platform Google Colab . . . . .	28
<b>5</b>	<b>Kesimpulan dan Saran</b>	<b>32</b>
5.1	Kesimpulan . . . . .	32
5.2	Saran . . . . .	32
<b>A</b>	<b>Source Code</b>	<b>36</b>

# Bab 1

## Pendahuluan

### 1.1 Latar Belakang

Kesehatan adalah salah satu pilar utama dalam kehidupan manusia, dengan penyakit kardiovaskular menjadi salah satu ancaman terbesar bagi kesejahteraan masyarakat di seluruh dunia. Setiap tahunnya, lebih dari 17 juta jiwa meninggal akibat penyakit kardiovaskular, menjadikannya sebagai penyebab utama kematian global. Di Indonesia, beban penyakit ini juga sangat signifikan. Berdasarkan data Riset Kesehatan Dasar (Riskesdas) tahun 2018, sekitar 1,5 persen dari populasi, atau lebih dari 2,7 juta orang, menderita penyakit jantung. Bahkan, laporan dari *Global Burden of Disease* tahun 2020 menunjukkan bahwa penyakit kardiovaskular menyebabkan sekitar 651.481 kematian per tahun di Indonesia. Gagal jantung, salah satu jenis penyakit kardiovaskular yang serius, terjadi ketika jantung tidak mampu memompa darah dengan cukup untuk memenuhi kebutuhan tubuh. Penyakit ini menjadi penyebab kematian kedua setelah stroke di Indonesia, dengan prevalensi yang diperkirakan mencapai lebih dari 1 juta orang (Chicco D. et al., 2020).

Dalam beberapa tahun terakhir, kemajuan teknologi kesehatan, termasuk penggunaan rekam medis elektronik (*Electronic Health Records/EHR*), telah menyediakan data berharga yang mencakup informasi seperti gejala, karakteristik tubuh, dan hasil uji laboratorium. Data ini memberikan peluang besar untuk analisis yang lebih mendalam guna mengidentifikasi pola dan hubungan yang sulit ditemukan melalui metode konvensional. Namun, meskipun telah banyak kemajuan dalam diagnosis dan pengobatan gagal jantung, tantangan tetap ada dalam memahami pola data pasien yang kompleks. Analisis data yang cermat diperlukan untuk membantu tenaga medis memahami faktor-faktor yang berkontribusi pada prognosis pasien.

Dalam konteks ini, penerapan metode *machine learning* seperti analisis *clustering* dengan algoritma K-Means menjadi relevan. Algoritma ini dapat mengelompokkan pasien berdasarkan karakteristik tertentu untuk mengidentifikasi pola-pola tersembunyi yang dapat mendukung pengambilan keputusan medis. Pemilihan data gagal jantung untuk penelitian ini didasarkan pada beberapa alasan.

1. Penyakit ini memiliki dampak yang besar terhadap mortalitas dan morbiditas, sehingga analisis yang lebih mendalam sangat diperlukan untuk mendukung upaya pencegahan dan pengobatan.

2. Dataset ini mencakup variabel-variabel yang relevan secara klinis, seperti usia, jenis kelamin, riwayat penyakit (anemia, diabetes, hipertensi), hasil uji laboratorium (kreatinin fosfokinase, kadar kreatinin serum, kadar natrium serum), hingga waktu penanganan, yang kesemuanya berkontribusi pada prognosis pasien gagal jantung.
3. Dataset ini memberikan peluang untuk mengeksplorasi bagaimana teknik *machine learning* dapat diterapkan dalam konteks medis untuk mengatasi tantangan interpretasi data yang kompleks.

Berbagai penelitian terdahulu menunjukkan relevansi penerapan *machine learning* untuk analisis data gagal jantung. Zhang et al. (2022) melakukan penelitian tentang prediksi prognosis pada pasien lanjut usia dengan sepsis menggunakan random survival forest, sementara Ali M., Al-Doori V., et al. (2023) mengusulkan pendekatan *machine learning* untuk analisis faktor risiko dan prediksi kelangsungan hidup pasien gagal jantung. Khanna D. et al. (2020) juga melakukan studi komparatif teknik klasifikasi (SVM, regresi logistik, dan jaringan saraf) untuk memprediksi prevalensi penyakit jantung.

Penelitian ini bertujuan untuk menerapkan algoritma K-Means pada dataset gagal jantung guna mengidentifikasi pola-pola tersembunyi yang dapat membantu pengambilan keputusan klinis. Dengan pendekatan ini, penelitian berjudul "Analisis *Clustering* Menggunakan Algoritma K-Means pada Dataset *Heart Failure Clinical Record*" diharapkan dapat memberikan kontribusi signifikan dalam analisis data pasien gagal jantung dan memperluas pemanfaatan machine learning di bidang kesehatan.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana penerapan algoritma K-Means *clustering* dalam mengelompokkan data pasien gagal jantung berdasarkan karakteristik klinis dan laboratorium?
2. Apa saja pola-pola atau kelompok utama yang dapat diidentifikasi dari hasil pengelompokan pasien gagal jantung menggunakan algoritma K-Means?
3. Apakah variabel-variabel seperti usia, jenis kelamin, anemia, kadar enzim kreatinin fosfokinase, penderita diabetes, fraksi ejeksi, penderita hipertensi, trombosit, kadar kreatinin serum, kadar natrium serum, riwayat merokok dan waktu penanganan, memiliki kontribusi signifikan dalam menentukan pengelompokan pasien?
4. Bagaimana hasil pengelompokan menggunakan algoritma K-Means dapat membantu pengambilan keputusan klinis dalam pengelolaan pasien gagal jantung?

## 1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Menerapkan algoritma K-Means *clustering* untuk mengelompokkan data pasien gagal jantung berdasarkan karakteristik klinis dan hasil laboratorium yang tersedia.
2. Mengidentifikasi pola-pola atau kelompok utama dari hasil pengelompokan pasien gagal jantung menggunakan algoritma K-Means.
3. Menganalisis kontribusi variabel-variabel seperti usia, jenis kelamin, anemia, kadar enzim kreatinin fosfokinase, penderita diabetes, fraksi ejeksi, penderita hipertensi, trombosit, kadar kreatinin serum, kadar natrium serum, riwayat merokok dan waktu penanganan dalam menentukan hasil pengelompokan pasien.
4. Mengevaluasi bagaimana hasil pengelompokan dengan algoritma K-Means dapat digunakan untuk mendukung pengambilan keputusan klinis dalam pengelolaan pasien gagal jantung.

## 1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

### 1. Manfaat Teoritis:

#### (a) Kontribusi pada Pengembangan Ilmu Pengetahuan

Penelitian ini memberikan kontribusi yang signifikan dalam pengembangan ilmu pengetahuan di bidang analisis data kesehatan, terutama dalam penerapan machine learning. Khususnya, metode K-Means *clustering* memungkinkan identifikasi pola-pola tersembunyi dalam data klinis pasien gagal jantung. Dengan cara ini, penelitian ini membantu dalam memahami karakteristik pasien berdasarkan berbagai faktor seperti usia, jenis kelamin, gejala, dan riwayat medis, yang berpotensi meningkatkan pengelompokan data pasien.

Manfaat teoritis ini dapat diukur melalui peningkatan akurasi dan relevansi pengelompokan dibandingkan dengan metode tradisional. Selain itu, kontribusi ini akan tercermin dalam publikasi ilmiah yang mengenalkan pendekatan baru ini kepada komunitas peneliti, serta peningkatan pemahaman mengenai pola-pola kesehatan yang dapat memperkaya literatur tentang gagal jantung.

#### (b) Referensi untuk Penelitian Selanjutnya

Penelitian ini dapat menjadi dasar bagi studi lanjutan yang menggunakan algoritma machine learning lainnya, seperti Random Forest atau Support Vector Machines (SVM), yang bertujuan untuk meningkatkan akurasi pengelompokan atau prediksi dalam data medis. Selain itu, penelitian ini membuka peluang untuk mengeksplorasi variabel tambahan, seperti faktor

genetik atau lingkungan, yang dapat memperdalam pemahaman tentang faktor-faktor yang mempengaruhi pasien gagal jantung.

Manfaat ini dapat diukur dari banyaknya penelitian yang merujuk pada penelitian ini dan menerapkan metode atau hasil yang serupa, serta keberhasilan penerapan variabel baru atau metode lain dalam meningkatkan kualitas hasil penelitian lebih lanjut.

## 2. Manfaat Praktis:

### (a) **Bagi Institusi Medis**

Penelitian ini memiliki manfaat praktis yang sangat penting bagi institusi medis, seperti rumah sakit atau klinik. Dengan mengidentifikasi pola-pola pasien gagal jantung berdasarkan pengelompokan data, rumah sakit dapat merancang strategi perawatan yang lebih tepat dan efisien. Misalnya, pengelompokan pasien berdasarkan tingkat keparahan dapat membantu institusi medis untuk menentukan alokasi sumber daya secara lebih efisien, seperti memberikan perhatian lebih kepada pasien yang lebih serius dan mengelola pasien dengan risiko lebih rendah dengan pendekatan yang lebih sederhana.

Manfaat praktis ini dapat diukur dengan mengamati peningkatan kualitas perawatan, pengurangan biaya medis, dan penurunan angka rawat inap berulang. Evaluasi dapat dilakukan dengan mengukur waktu perawatan, tingkat kepuasan pasien, dan efisiensi biaya.

### (b) **Bahan Pertimbangan dalam Pengambilan Keputusan Medis**

Hasil dari penelitian ini dapat digunakan oleh tenaga medis untuk membuat keputusan yang lebih tepat terkait diagnosis dan manajemen risiko. Misalnya, pengelompokan pasien yang lebih akurat dapat membantu dokter menentukan tingkat keparahan penyakit dan memilih terapi yang lebih sesuai. Pasien dengan risiko tinggi bisa mendapatkan pengobatan yang lebih agresif, sedangkan pasien dengan risiko rendah dapat menjalani pendekatan pencegahan yang lebih ringan.

Manfaat ini bisa diukur dengan peningkatan akurasi diagnosis dan pengurangan waktu yang diperlukan untuk mencapai keputusan yang tepat. Selain itu, manfaat ini dapat terlihat dari pengurangan komplikasi atau peningkatan outcome pasien yang terklaster dengan cara yang lebih efektif dan efisien.

## Bab 2

# Tinjauan Pustaka

### 2.1 Gagal Jantung (Heart Failure)

Gagal jantung adalah kondisi medis kronis di mana jantung tidak mampu memompa darah secara efisien untuk memenuhi kebutuhan metabolisme tubuh. Kondisi ini dapat disebabkan oleh kelemahan otot jantung atau kekakuan dinding jantung yang mengurangi kapasitas pompa darahnya (Doifode et al., 2024). Menurut *American Heart Association* (AHA), gagal jantung adalah salah satu penyebab utama kematian dan kesakitan di seluruh dunia yang memerlukan penanganan segera dan menyeluruh untuk meningkatkan prognosis pasien.

Menurut *World Health Organization* (WHO), beberapa penyebab utama gagal jantung meliputi penyakit arteri koroner, hipertensi, serangan jantung (*myocardial infarction*), kelainan katup jantung, dan kardiomiopati. Penyakit arteri koroner adalah penyebab yang paling umum, di mana aliran darah ke otot jantung terganggu akibat penyempitan atau penyumbatan arteri. Selain itu, hipertensi yang tidak terkontrol menyebabkan beban kerja jantung meningkat, yang akhirnya melemahkan otot jantung. Kerusakan otot jantung akibat serangan jantung atau kelainan struktural pada katup jantung juga turut berkontribusi terhadap perkembangan gagal jantung.

Gagal jantung ditandai oleh gejala seperti sesak napas, kelelahan, pembengkakan di kaki atau perut akibat retensi cairan, dan ketidakmampuan melakukan aktivitas fisik yang berat. Diagnosisnya melibatkan kombinasi wawancara klinis, pemeriksaan fisik, dan tes diagnostik. Tes seperti ekokardiografi dapat memberikan gambaran tentang fungsi jantung, sementara kadar natriuretic peptide (BNP atau NT-proBNP) dalam darah sering digunakan untuk mengonfirmasi gagal jantung. Elektrokardiogram (EKG) juga membantu mendeteksi irama jantung yang abnormal (Halvorsen et al., 2022).

Pengobatan gagal jantung mencakup terapi medis seperti diuretik untuk mengurangi retensi cairan, ACE inhibitors untuk melebarkan pembuluh darah, dan beta-blockers untuk mengurangi tekanan darah serta memperbaiki efisiensi jantung. Pada kasus yang lebih parah, perangkat medis seperti defibrillator implan atau alat bantu ventricular dapat digunakan. Operasi transplantasi jantung merupakan opsi terakhir bagi pasien yang tidak merespons pengobatan lainnya. Selain itu, rehabilitasi jantung yang melibatkan latihan fisik dan edukasi pasien sangat penting untuk meningkatkan kualitas hidup.

Gagal jantung memengaruhi lebih dari 64 juta orang di seluruh dunia, dengan tingkat kejadian yang terus meningkat, terutama di kalangan populasi lansia. Di Amerika Serikat saja, lebih dari 6,2 juta orang hidup dengan kondisi ini. Prevalensi gagal jantung sering dikaitkan dengan meningkatnya faktor risiko seperti hipertensi, obesitas, dan diabetes (Savarese et al., 2022). Oleh karena itu, deteksi dini dan pengobatan yang tepat sangat penting untuk mencegah perkembangan penyakit.

## 2.2 Dataset *Heart Failure Clinical Record*

Dataset Heart Failure Clinical Records adalah kumpulan data klinis yang dirancang untuk mendukung analisis prediktif dan penelitian terkait kondisi gagal jantung. Dataset ini mencakup data dari 299 pasien yang mengalami gagal jantung, dengan informasi yang mencakup 13 atribut medis dan demografis, seperti usia, kadar kreatinin serum, fraksi ejeksi ventrikel kiri (*ejection fraction*), tekanan darah tinggi, dan kebiasaan merokok (Davide Chicco et al., 2020). Dataset ini berasal dari *Faisalabad Institute of Cardiology* dan *Allied Hospital*, Faisalabad, Pakistan, dan tersedia secara luas melalui *UCI Machine Learning Repository*, sebuah repositori terkemuka yang sering digunakan oleh komunitas penelitian global.

Dataset ini telah banyak digunakan dalam eksperimen *machine learning*, khususnya untuk klusterisasi, karena beberapa alasan utama, yaitu :

1. Dataset ini memiliki ukuran yang relatif kecil, dengan dimensi 299 sampel dan 13 fitur, sehingga ideal untuk penelitian awal dan eksplorasi metode *machine learning*.
2. Atribut yang disediakan mencakup data numerik dan biner, yang memungkinkan penerapan berbagai algoritma klusterisasi seperti K-Means atau *Hierarchical Clustering*.
3. Relevansi klinisnya sangat tinggi, mengingat tingginya tingkat kematian akibat gagal jantung.

Dataset ini dapat membantu segmentasi pasien menjadi kelompok risiko berdasarkan karakteristik medis mereka, yang pada akhirnya mendukung pengambilan keputusan klinis untuk intervensi yang lebih efektif. Selain itu, dataset ini memberikan peluang untuk mengeksplorasi pola tersembunyi yang dapat meningkatkan pemahaman tentang faktor risiko gagal jantung.

Dataset ini telah digunakan dalam berbagai penelitian, termasuk aplikasi metode prediksi seperti klasifikasi dan eksplorasi pola menggunakan algoritma klusterisasi. Dengan keberagaman atribut dan relevansi medis yang tinggi, dataset ini menjadi sumber data yang berharga dalam penelitian berbasis *machine learning* untuk pengelolaan pasien gagal jantung.



## 2.3 Metode *Clustering*

Metode *clustering* adalah salah satu teknik utama dalam analisis data yang bertujuan untuk mengelompokkan data ke dalam kelompok-kelompok (klaster) berdasarkan kesamaan tertentu. Dalam *clustering*, objek-objek dalam kelompok yang sama memiliki karakteristik yang lebih mirip satu sama lain dibandingkan dengan objek-objek di kelompok lain. Teknik ini termasuk dalam kategori *unsupervised learning*, yang berarti tidak memerlukan label atau target variabel untuk melatih modelnya.

### Jenis-Jenis Metode *Clustering*

#### 1. *Clustering* Berbasis Partisi

Metode ini mempartisi dataset menjadi beberapa kelompok, biasanya dengan meminimalkan jarak antara data dalam klaster. Contoh populer adalah algoritma K-Means, yang membagi data ke dalam sejumlah  $k$  cluster berdasarkan centroid. Algoritma ini iteratif dan bertujuan untuk meminimalkan *intra-cluster variance*.

#### 2. *Clustering* Berbasis Hirarki

Metode ini menciptakan struktur hierarki dalam bentuk pohon (dendrogram). Data diorganisasikan mulai dari satu klaster besar hingga klaster kecil, atau sebaliknya. *Clustering* hierarki dibagi menjadi dua pendekatan, yaitu agglomerative (pendekatan bottom-up) dan divisive (pendekatan top-down).

#### 3. *Clustering* Berbasis Kepadatan

Metode ini mengelompokkan data berdasarkan kepadatan titik-titik data dalam ruang tertentu. Algoritma seperti DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) sering digunakan untuk mendeteksi klaster dengan bentuk arbitrer serta menangani data dengan outlier (Ester et al., 1996).

#### 4. *Clustering* Berbasis Model

Metode ini mengasumsikan bahwa data dihasilkan dari distribusi tertentu, seperti distribusi Gaussian, dan mencoba untuk menemukan parameter model yang paling cocok untuk mendeskripsikan data. Contoh yang sering digunakan adalah *Gaussian Mixture Models* (GMM).

### Aplikasi dan Manfaat *Clustering*

*Clustering* digunakan di berbagai bidang, termasuk:

- Kesehatan: Mengelompokkan pasien berdasarkan gejala atau faktor risiko untuk membantu diagnosis atau rencana perawatan yang lebih personal.
- Pemasaran: Mengelompokkan pelanggan berdasarkan preferensi atau perilaku untuk strategi pemasaran yang ditargetkan.
- Biologi: Mengelompokkan gen atau protein berdasarkan fungsi atau ekspresi mereka.
- Analisis Sosial: Menganalisis jaringan sosial untuk menemukan komunitas atau kelompok yang saling berhubungan (Jain, AK et al., 2000).

### ***Clustering pada Dataset Heart Failure Clinical Records***

Dalam konteks dataset *Heart Failure Clinical Records*, metode *clustering* seperti K-Means sangat relevan. Algoritma ini digunakan untuk mengelompokkan pasien berdasarkan atribut klinis seperti usia, kadar natrium serum, atau fraksi ejeksi. Tujuan utamanya adalah untuk menemukan pola tersembunyi dalam data pasien yang dapat membantu mengidentifikasi kelompok risiko tinggi dan rendah. Informasi ini dapat digunakan oleh tenaga medis untuk menentukan prioritas perawatan atau intervensi yang lebih tepat. Dengan demikian, *clustering* berfungsi sebagai alat penting untuk mendukung pengambilan keputusan berbasis data dalam manajemen gagal jantung (Davide Chicco et al., 2020).

## **2.4 Algoritma K-Means dalam Analisis Data Medis**

Algoritma K-Means adalah metode clustering berbasis partisi yang membagi data menjadi beberapa kelompok (klaster) berdasarkan kesamaan antar data. Setiap klaster ditentukan oleh centroid, yang merupakan rata-rata dari semua data dalam klaster tersebut. Proses iteratif dilakukan untuk meminimalkan variasi internal dalam klaster dan memaksimalkan perbedaan antar klaster. Tujuan utama adalah mengurangi jarak total kuadrat antara data dan centroid klaster (fungsi SSE—Sum of Squared Errors).

Dalam analisis medis, K-Means sering digunakan untuk memahami pola data pasien dan mendukung pengambilan keputusan klinis. Contoh penggunaannya adalah untuk membagi pasien berdasarkan karakteristik seperti usia, kadar gula darah, tekanan darah, atau hasil lab lainnya, mengidentifikasi kelompok pasien dengan risiko tinggi, misalnya gagal jantung atau diabetes, memprediksi bagaimana kelompok tertentu merespons pengobatan tertentu.

Kumar et al. (2020) menunjukkan bagaimana K-Means membantu mengelompokkan pasien gagal jantung berdasarkan parameter fisiologis seperti Ejection fraction, Kadar natrium serum, usia dan waktu penanganan. Hal ini memungkinkan tenaga medis untuk mengidentifikasi kelompok risiko tinggi dan mengambil langkah preventif lebih awal.

Pada dataset *Heart Failure Clinical Records*, K-Means dapat digunakan untuk mengelompokkan pasien berdasarkan usia, kadar natrium serum, dan fraksi ejeksi untuk memahami risiko gagal jantung. Serta dapat mengidentifikasi pola pasien dengan risiko tinggi untuk komplikasi medis.

## Bab 3

# Metode Penelitian

### 3.1 Deskripsi Penelitian dan Dataset

Penelitian ini bertujuan untuk mengelompokkan pasien gagal jantung berdasarkan pola data klinis mereka menggunakan algoritma K-Means Clustering. Dataset yang digunakan adalah Heart Failure Clinical Records yang diambil dari UCI Machine Learning Repository. Dataset ini berisi 299 entri pasien dengan 1 variabel target (kematian akibat gagal jantung) dan 12 fitur klinis, antara lain usia, tekanan darah, kadar kolesterol, fraksi ejeksi, kadar natrium serum, kadar kreatinin serum, serta status medis lainnya seperti riwayat diabetes dan hipertensi.

Proses analisis dimulai dengan eksplorasi awal data untuk memahami distribusi setiap fitur, mendeteksi adanya outlier, dan mengidentifikasi pola-pola awal dalam dataset. Tahap ini bertujuan untuk memastikan bahwa data yang digunakan memenuhi kualitas yang diperlukan untuk analisis lebih lanjut. Selain itu, eksplorasi data juga mencakup langkah-langkah pembersihan data, seperti menangani nilai yang hilang atau tidak konsisten, serta visualisasi data untuk mendapatkan wawasan mengenai hubungan antar fitur yang ada sebelum algoritma K-Means diterapkan.

### 3.2 Variabel Penelitian

Variabel adalah sesuatu yang dapat diukur atau diamati dan memiliki nilai yang bervariasi.

1. Variabel independen (bebas): Karakteristik klinis dan laboratorium pasien yang mencakup:
  - Usia
  - Kadar natrium serum
  - Ejection fraction
  - Serum creatinine
2. Variabel dependen (terikat): Kelompok hasil *clustering* yang dihasilkan oleh algoritma K-Means. Kelompok ini mencerminkan pola pengelompokan pasien berdasarkan tingkat risiko atau karakteristik klinis mereka.

3. Variabel moderator: Jenis kelamin pasien dan riwayat penyakit tertentu (seperti hipertensi atau diabetes). Variabel ini dapat memengaruhi hubungan antara karakteristik klinis (variabel independen) dan hasil *clustering* (variabel dependen.)
4. Variabel kontrol: durasi pengamatan (time) dan unit pengukuran data klinis dan laboratorium.

### 3.3 Instrumen Penelitian

Instrumen penelitian adalah alat yang digunakan untuk mengumpulkan, mengolah, dan menganalisis data pada penelitian ini. Dalam konteks penerapan algoritma K-Means *clustering* pada dataset Heart Failure Clinical Record, instrumen yang digunakan meliputi:

1. Dokumentasi Dataset Penelitian

Dataset yang digunakan adalah Heart Failure Clinical Record, yang berisi data pasien dengan variabel-variabel klinis dan laboratorium seperti:

- Usia  
Usia adalah salah satu faktor paling penting dalam menganalisis risiko gagal jantung. Secara alami, seiring bertambahnya usia, tubuh manusia mengalami penurunan dalam fungsi organ-organ vital, termasuk jantung. Pada usia yang lebih tua, dinding jantung cenderung menjadi lebih kaku, dan kemampuan untuk memompa darah dengan efisien juga menurun. Hal ini memicu peningkatan tekanan darah, penurunan fungsi sirkulasi darah, dan ketidakseimbangan cairan, yang semuanya menjadi faktor yang memperburuk kondisi gagal jantung. Penurunan kapasitas jantung untuk memompa darah dengan baik pada usia tua meningkatkan kemungkinan terjadinya gagal jantung pada individu tersebut.
- Kadar Natrium Serum  
Natrium merupakan elektrolit yang sangat penting dalam menjaga keseimbangan cairan dan volume darah dalam tubuh. Kadar natrium serum yang rendah (hiponatremia) sering menjadi indikator utama dalam pasien dengan gagal jantung. Hiponatremia sering terjadi pada pasien gagal jantung karena akumulasi cairan yang berlebihan dalam tubuh yang tidak bisa dikeluarkan dengan efektif oleh ginjal. Ketidakseimbangan ini dapat memengaruhi fungsi jantung dan menyebabkan gejala-gejala seperti sesak napas dan pembengkakan (edema). Oleh karena itu, kadar natrium serum yang rendah seringkali menunjukkan prognosis yang lebih buruk pada pasien gagal jantung, karena meningkatkan beban pada jantung dan menurunkan kapasitasnya untuk memompa darah.
- Fraksi Ejeksi  
Fraksi ejeksi (ejection fraction) adalah persentase darah yang dipompa keluar dari ventrikel kiri jantung pada setiap detak jantung. Nilai normal fraksi ejeksi adalah sekitar 55 persen -70 persen . Sebaliknya, jika fraksi ejeksi berada di bawah nilai tersebut, maka kondisi jantung dikategorikan dalam risiko gagal jantung. Fraksi ejeksi yang rendah menunjukkan bahwa jantung tidak memompa darah dengan baik, yang merupakan ciri utama dari gagal

jantung. Dengan kata lain, semakin rendah fraksi ejeksi, semakin besar kemungkinan pasien tersebut mengalami penurunan fungsi jantung dan lebih rentan terhadap kondisi gagal jantung yang lebih parah. Pemantauan dan pengobatan yang tepat untuk meningkatkan fraksi ejeksi sangat penting dalam mengelola gagal jantung.

- Kadar Kreatinin Serum

Kreatinin adalah produk limbah yang dihasilkan oleh otot dan disaring oleh ginjal. Kadar kreatinin dalam darah memberikan indikasi yang baik mengenai fungsi ginjal. Kadar kreatinin yang tinggi sering kali menandakan adanya gangguan pada ginjal, yang dapat terjadi bersamaan dengan gagal jantung. Gangguan fungsi ginjal pada pasien gagal jantung sering kali mengarah pada akumulasi cairan yang lebih tinggi dalam tubuh, yang pada gilirannya memperburuk gejala gagal jantung, seperti sesak napas dan pembengkakan. Peningkatan kadar kreatinin dalam darah dapat menunjukkan bahwa tubuh pasien gagal jantung mengalami kesulitan dalam mengeluarkan limbah, yang mengindikasikan penurunan fungsi ginjal akibat kerusakan yang terkait dengan gagal jantung.

- Kadar Enzim Kreatinin Fosfokinase

Kreatinin fosfokinase (CPK) adalah enzim yang ditemukan dalam berbagai jaringan tubuh, termasuk otot jantung. Ketika otot jantung mengalami kerusakan, seperti yang terjadi pada serangan jantung atau gagal jantung, kadar CPK dalam darah akan meningkat. Peningkatan kadar CPK bisa menunjukkan adanya kerusakan pada jaringan jantung atau bahkan infark miokard (serangan jantung). Oleh karena itu, CPK sering digunakan dalam pengukuran tingkat kerusakan jantung pada pasien gagal jantung dan memberikan wawasan penting mengenai keparahan kondisi jantung pasien.

- Riwayat Hipertensi

Hipertensi atau tekanan darah tinggi adalah salah satu faktor utama yang dapat menyebabkan dan memperburuk kondisi gagal jantung. Tekanan darah tinggi menyebabkan pembuluh darah menjadi kaku dan sempit, yang meningkatkan beban pada jantung. Jantung harus bekerja lebih keras untuk memompa darah, yang lama kelamaan dapat menyebabkan kerusakan pada otot jantung, meningkatkan risiko gagal jantung. Oleh karena itu, hipertensi yang tidak terkontrol merupakan salah satu faktor risiko utama yang harus diwaspadai dalam pengelolaan pasien gagal jantung.

- Jenis Kelamin

Jenis kelamin memiliki pengaruh yang signifikan terhadap kesehatan jantung dan risiko gagal jantung. Pria lebih cenderung mengembangkan gagal jantung pada usia yang lebih muda, sedangkan wanita sering mengembangkan kondisi ini setelah menopause, ketika kadar hormon estrogen menurun. Selain itu, wanita cenderung mengalami jenis gagal jantung yang lebih kompleks dan sering kali berhubungan dengan gejala yang lebih parah. Penanganan dan pendekatan pengobatan bisa berbeda antara pria dan wanita, sehingga penting untuk mempertimbangkan jenis kelamin dalam merencanakan perawatan untuk pasien gagal jantung.

- Trombosit

Trombosit adalah sel darah yang berfungsi dalam proses pembekuan da-

rah. Ketika trombosit berada dalam jumlah yang tidak normal, baik terlalu tinggi maupun rendah, hal ini dapat memengaruhi sirkulasi darah dan pembekuan. Pada pasien gagal jantung, gangguan dalam sirkulasi darah dapat menyebabkan peningkatan jumlah trombosit yang berlebihan, yang meningkatkan risiko pembekuan darah dan stroke. Sebaliknya, jumlah trombosit yang rendah dapat meningkatkan risiko pendarahan.

- **Diabetes**  
Diabetes mellitus adalah kondisi yang mempengaruhi cara tubuh mengelola gula darah. Pasien dengan diabetes memiliki risiko yang lebih tinggi untuk mengalami kerusakan pembuluh darah dan jantung, yang dapat mengarah pada gagal jantung. Diabetes memperburuk kerja jantung karena meningkatkan resistensi insulin dan menyebabkan penumpukan lemak di pembuluh darah. Pada pasien yang sudah memiliki gagal jantung, diabetes sering kali memperburuk kondisi dan memperpendek harapan hidup mereka.
- **Anemia**  
Anemia adalah kondisi di mana tubuh kekurangan sel darah merah yang sehat untuk membawa oksigen ke seluruh tubuh. Pada pasien gagal jantung, anemia dapat memperburuk gejala seperti kelelahan, sesak napas, dan penurunan kemampuan fisik. Kekurangan oksigen pada jaringan tubuh dapat memperburuk beban kerja jantung, yang akhirnya memperburuk kondisi gagal jantung. Oleh karena itu, pengelolaan anemia sangat penting pada pasien gagal jantung untuk meningkatkan kualitas hidup mereka.
- **Kebiasaan Merokok**  
Merokok adalah salah satu faktor yang dapat memperburuk kesehatan jantung. Nikotin dan bahan kimia lain dalam rokok dapat merusak dinding pembuluh darah, meningkatkan tekanan darah, serta mengurangi aliran darah ke organ vital, termasuk jantung. Kebiasaan merokok juga dapat meningkatkan pembentukan plak di arteri (aterosklerosis), yang memperburuk kondisi jantung dan meningkatkan risiko gagal jantung.
- **Waktu Penanganan**  
Waktu penanganan mengacu pada berapa lama pasien menerima pengobatan sejak didiagnosis dengan gagal jantung. Penanganan yang lebih cepat dapat mengurangi beban pada jantung, mencegah komplikasi lebih lanjut, dan memperpanjang harapan hidup pasien. Keterlambatan dalam penanganan dapat memperburuk gejala dan meningkatkan risiko kematian.

## 2. Software dan Alat Analisis

- **Google Colab:** Digunakan sebagai platform untuk analisis data berbasis Python. Platform ini dipilih karena kemudahan akses, kemampuan untuk berbagi kode, dan integrasi dengan pustaka Python yang relevan untuk *machine learning*.
- **Pustaka python:**
  - Pandas untuk pengelolaan dan manipulasi data.
  - NumPy untuk komputasi numerik.
  - Scikit-learn untuk implementasi algoritma K-Means dan evaluasi hasil *clustering*.

## 3.4 Teknik Pengumpulan Data

Penelitian ini menggunakan teknik pengumpulan data berbasis **studi literatur** sebagai pendekatan utama. Pendekatan ini bertujuan untuk mengumpulkan informasi dari berbagai penelitian terdahulu yang relevan, termasuk kajian tentang algoritma K-Means, teknik *clustering*, serta aplikasi pembelajaran mesin dalam analisis dan pengelolaan pasien gagal jantung. Dengan memanfaatkan studi literatur, penelitian ini dapat memperoleh wawasan yang telah teruji, serta membangun fondasi yang lebih komprehensif dalam melakukan analisis terhadap masalah yang dihadapi.

Agar data yang digunakan dalam penelitian ini relevan dan berkualitas, diterapkan kriteria inklusi dan eksklusi yang dirancang dengan cermat untuk menyaring informasi dari dataset. Kriteria inklusi mencakup data pasien yang telah didiagnosis dengan gagal jantung dan tercatat dalam dataset Heart Failure Clinical Record. Variabel penting yang harus tersedia dalam dataset meliputi usia, kadar natrium serum, fraksi ejeksi, dan kadar kreatinin serum. Penerapan kriteria inklusi ini bertujuan untuk memastikan bahwa dataset yang digunakan memenuhi syarat untuk mendukung analisis yang valid. Dalam penelitian ini, seluruh data dalam dataset telah memenuhi kriteria inklusi, sehingga tidak ada kasus anomali seperti usia pasien yang bernilai negatif atau data yang tidak relevan dengan penelitian.

Sebaliknya, kriteria eksklusi diterapkan untuk menghindari penggunaan data yang dapat mengurangi keakuratan hasil analisis klastering. Misalnya, data dengan nilai biner seperti anemia, diabetes, hipertensi, jenis kelamin, dan kebiasaan merokok dikeluarkan dari analisis karena tidak berkontribusi secara signifikan terhadap proses klastering berbasis algoritma K-Means. Selain itu, variabel seperti tekanan darah dapat memengaruhi hasil klastering karena cenderung menghasilkan interpretasi yang kurang stabil dan dapat menambah variabilitas yang tidak diinginkan dalam pengelompokan data. Dengan menerapkan kriteria eksklusi ini, hanya data yang relevan dan mendukung analisis klastering yang akan disertakan, sehingga algoritma K-Means dapat bekerja secara optimal dan menghasilkan klaster yang lebih akurat.

Jika terdapat data yang hilang (*missing values*), dua pendekatan dapat digunakan untuk mengatasinya, yaitu metode imputasi dan metode eksklusi. Metode imputasi digunakan untuk mengisi nilai yang hilang dalam dataset dengan cara memperkirakan nilai tersebut berdasarkan pola atau hubungan dengan variabel lain dalam dataset. Teknik imputasi seperti pengisian dengan nilai rata-rata, median, atau menggunakan model prediktif dapat memastikan bahwa dataset tetap utuh tanpa kehilangan informasi penting. Di sisi lain, metode eksklusi diterapkan untuk menghapus data yang tidak lengkap atau bermasalah, terutama jika data yang hilang terlalu signifikan atau tidak dapat diimputasi dengan akurat. Dalam penelitian ini, metode eksklusi lebih diutamakan apabila nilai yang hilang pada variabel penting cukup besar, karena penghapusan data tersebut dapat memastikan bahwa hanya data yang relevan dan lengkap yang digunakan dalam analisis.

Dengan pengelolaan data yang sistematis, penerapan kriteria inklusi dan eksklusi yang ketat, serta penggunaan metode imputasi atau eksklusi yang sesuai, penelitian ini diharapkan dapat menghasilkan analisis yang valid, relevan, dan mendalam. Hal ini akan mendukung pengambilan keputusan berbasis data dalam penanganan dan pengelolaan kasus gagal jantung secara lebih efektif.

### 3.5 Teknik Analisis Data

Penelitian ini menggunakan pendekatan kuantitatif untuk menganalisis data dalam dataset Heart Failure Clinical Record. Berikut adalah tahapan dan teknik yang diterapkan:

#### 1. *Preprocessing Data*

- **Pembersihan Data.** Data diperiksa untuk mendeteksi missing values, duplikasi, dan anomali:
  - **Missing Values.** Pada missing values dapat ditangani dengan metode imputasi (dengan mengisi nilai hilang dengan rata-rata, median, atau lainnya) atau eksklusi (menghapus data tidak lengkap jika signifikan memengaruhi analisis)
  - **Duplikasi.** Data ganda yang dihapus menggunakan fungsi seperti ".duplicated()" untuk menghindari bias.
  - **Anomali.** Nilai yang menyimpang jauh dari pola (outliers) ditangani menggunakan teknik seperti boxplot atau z-score.
- **Standarisasi Data.** Variabel dalam dataset distandarisasi menggunakan StandardScaler dari pustaka scikit-learn untuk memastikan semua variabel memiliki rata-rata 0 dan standar deviasi 1. Manfaatnya untuk meningkatkan kinerja algoritma berbasis jarak, seperti K-Means, dengan memastikan semua variabel memiliki skala yang sama sehingga analisis lebih akurat dan stabil.

#### 2. **Reduksi Dimensi dengan PCA (*Principal Component Analysis*)**

PCA digunakan untuk mengurangi dimensi data dengan memfokuskan analisis pada komponen utama yang menjelaskan variabilitas terbesar dalam dataset. Dengan menyederhanakan data yang penting, analisis dapat lebih efisien. Hasil transformasi PCA digunakan sebagai input untuk algoritma K-Means, memungkinkan proses klustering menjadi lebih optimal.

#### 3. **Menentukan Jumlah Kluster Optimal dengan Metode Elbow**

Untuk menentukan jumlah kluster optimal, digunakan **metode elbow**, yang menghitung nilai inertia (jumlah kuadrat jarak dalam kluster) untuk berbagai nilai k menggunakan perhitungan **WCSS** (Within-Custom Sum Of Squares). Dari grafik tersebut, terdapat **Elbow Point**, yaitu titik di mana penurunan WCSS mulai melambat, membentuk sudut seperti "siku", yang menunjukkan jumlah kluster yang ideal untuk analisis.

#### 4. **Clustering dengan Algoritma K-Means**

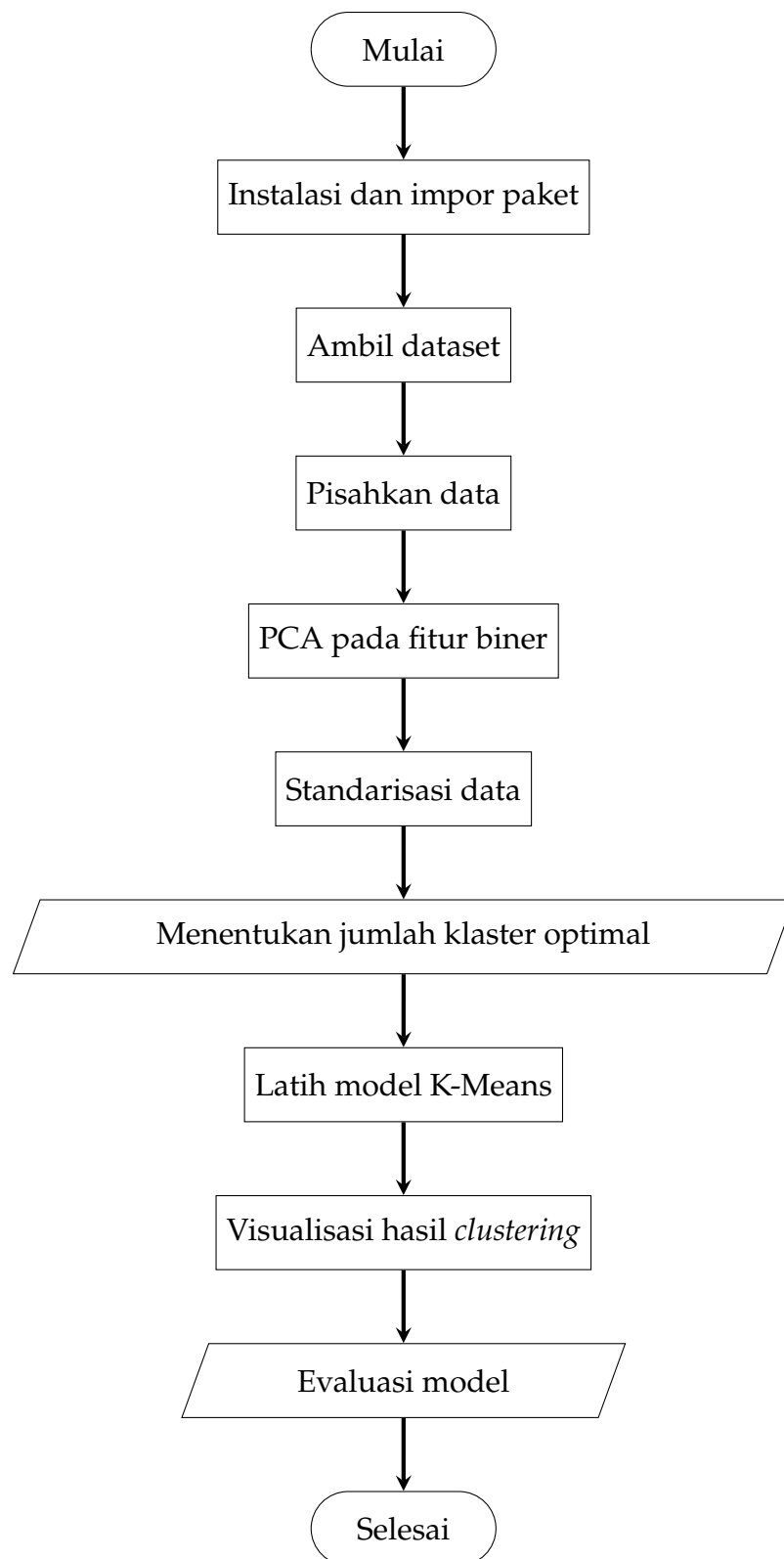
Setelah nilai k optimal ditentukan, algoritma K-Means diterapkan pada data yang telah direduksi dimensinya dengan PCA.

#### 5. **Visualisasi Hasil Clustering**

Hasil kluster divisualisasikan dalam dua dimensi untuk mempermudah interpretasi pola-pola yang muncul.



### 3.6 Algoritma Penelitian



Gambar 3.1: Alur Proses K-Means *Clustering*

# Bab 4

## Hasil dan Pembahasan

### 4.1 *Pre-Processing Data*

#### 4.1.1 *Pengecekan Missing Value*

Pengecekan terhadap data kosong (*missing values*) merupakan langkah krusial dalam memastikan kualitas dan keakuratan data yang digunakan dalam analisis. Dalam konteks analisis data, khususnya pada metode *clustering*, data yang lengkap dan konsisten sangat penting untuk menghasilkan pembagian kelompok atau klaster yang akurat. Oleh karena itu, pengecekan terhadap data kosong dilakukan untuk memastikan bahwa data yang diunduh atau dikumpulkan benar-benar lengkap dan tidak ada informasi yang hilang. Data kosong sering kali menjadi indikator adanya masalah dalam tahapan pengumpulan, penyimpanan, atau pengunduhan data, yang jika tidak ditangani dengan benar, dapat menyebabkan hasil analisis yang bias dan tidak akurat.

Pengecekan terhadap *missing value* sangat penting dalam tahapan *pre-processing* data. Tahapan ini bertujuan untuk memastikan bahwa dataset yang digunakan dalam analisis *clustering* benar-benar siap, tanpa adanya informasi yang hilang yang dapat mengganggu proses analisis. Pengecekan ini memungkinkan identifikasi variabel yang memiliki data yang tidak lengkap, yang jika dibiarkan tanpa penanganan yang tepat, dapat mempengaruhi keakuratan hasil analisis *clustering* yang dilakukan. Data yang tidak lengkap dapat menyebabkan algoritma *clustering*, seperti *K-means*, menghasilkan cluster yang tidak representatif atau bahkan menyesatkan.

Pengecekan terhadap *missing value* pada dataset dilakukan dengan menggunakan teknik eksplorasi data yang umum, yaitu dengan memeriksa jumlah dan persentase nilai yang hilang di setiap variabel dalam dataset. Langkah ini bertujuan untuk mengidentifikasi variabel-variabel yang mengandung data yang hilang, serta untuk mengetahui seberapa besar proporsi *missing value* pada setiap fitur dalam dataset.

Berikut pengecekan data kosong dengan menggunakan bantuan *platform* google colab berbasis python:



```
x.isnull().sum()
```

	0
age	0
anaemia	0
creatinine_phosphokinase	0
diabetes	0
ejection_fraction	0
high_blood_pressure	0
platelets	0
serum_creatinine	0
serum_sodium	0
sex	0
smoking	0
time	0

dtype: int64

Gambar 4.1: Pengecekan data kosong

Dari gambar di atas, dapat dilihat bahwa seluruh data dalam dataset telah terisi secara lengkap. Jadi dapat disimpulkan bahwa semua variabel dalam dataset tidak mengandung *missing value*. Hal ini menunjukkan bahwa tidak ada nilai yang hilang pada setiap variabel, yang berarti kita tidak perlu khawatir tentang potensi bias yang sering timbul akibat adanya data yang tidak lengkap. Dengan kondisi ini, analisis statistik yang dilakukan akan lebih akurat dan valid, karena tidak ada langkah tambahan yang diperlukan untuk menangani *missing value*, seperti imputasi atau penghapusan baris data. Oleh karena itu, kita dapat melanjutkan proses analisis lebih lanjut tanpa khawatir mengenai kualitas data yang hilang, sehingga hasil analisis yang diperoleh dapat lebih diandalkan dan representatif.

### 4.1.2 Reduksi Data

Proses reduksi data bertujuan untuk mengurangi jumlah fitur dalam dataset, sambil mempertahankan informasi yang penting dan relevan. Salah satu teknik reduksi data yang umum digunakan adalah *Principal Component Analysis* (PCA). Meskipun PCA umumnya diterapkan pada data numerik, ada beberapa situasi di mana PCA juga digunakan pada data biner, yaitu data yang hanya memiliki dua nilai, seperti 0 dan 1, yang sering ditemukan dalam pengolahan data kategorikal atau dalam data hasil pengkodean biner.

Peleburan data dengan PCA pada data biner berfungsi untuk mengurangi dimensi fitur biner tanpa mengorbankan informasi penting yang terkandung dalam data tersebut. Dalam konteks ini, PCA bertujuan untuk mengubah data biner menjadi kombinasi linier dari fitur-fitur biner yang ada, yang disebut sebagai komponen utama, yang dapat mewakili sebagian besar variansi dalam data.

PCA pada dasarnya adalah teknik yang digunakan untuk menemukan arah utama variansi dalam dataset dan mengubah data ke dalam koordinat baru yang disebut komponen utama. Dalam hal data numerik, variansi dihitung menggunakan matriks kovarians, dan komponen utama adalah kombinasi linier dari fitur-fitur asli yang memiliki variansi terbesar. Namun, pada data biner, penggunaan PCA lebih kompleks

karena data tersebut hanya berisi dua nilai (0 atau 1). Variansi dalam data biner tidak dapat dihitung dengan cara yang sama seperti pada data kontinu.

Berikut PCA dengan menggunakan bantuan platform google colab berbasis python:

```
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
# print(X)
feature_pca = ['anaemia', 'diabetes', 'high_blood_pressure', 'sex', 'smoking']
feature = []

for i in x:
    if i not in feature_pca:
        feature.append(i)

data_biner = x[feature_pca]
data_non_biner = x[feature]

print(data_biner)
print(data_non_biner)
# proses PCA
pca = PCA(n_components=1)
pca_data = pca.fit_transform(data_biner)

pca_df = pd.DataFrame(pca_data, columns=["PCA"])

print("Data PCA:")
print(pca_df)
```

Gambar 4.2: Program PCA

```
anaemia  diabetes  high_blood_pressure  sex  smoking
0         0         0                   1    1         0
1         0         0                   0    1         0
2         0         0                   0    1         1
3         1         0                   0    1         0
4         1         1                   0    0         0
..      ...      ...                   ...  ...      ...
294        0         1                   1    1         1
295        0         0                   0    0         0
296        0         1                   0    0         0
297        0         0                   0    1         1
298        0         0                   0    1         1

[299 rows x 5 columns]
age      creatinine_phosphokinase  ejection_fraction  platelets \
0      75.0                    582                   20  265000.00
1      55.0                   7861                   38  263358.03
2      65.0                    146                   20  162000.00
3      50.0                    111                   20  210000.00
4      65.0                    160                   20  327000.00
..      ...      ...                   ...  ...      ...
294    62.0                     61                   38  155000.00
295    55.0                   1820                   38  270000.00
296    45.0                   2060                   60  742000.00
297    45.0                   2413                   38  140000.00
298    50.0                    196                   45  395000.00
```

Gambar 4.3: Hasil Running Program PCA

```

serum_creatinine serum_sodium time
0      1.9      130      4
1      1.1      136      6
2      1.3      129      7
3      1.9      137      7
4      2.7      116      8
..      ...      ...      ...
294     1.1      143     270
295     1.2      139     271
296     0.8      138     278
297     1.4      140     280
298     1.6      136     285

[299 rows x 7 columns]
Data PCA:
      PCA
0  0.179355
1  0.361319
2  0.960308
3  0.100695
4 -0.903569
..      ...
294 0.410196
295 -0.274796
296 -0.642945
297 0.960308
298 0.960308

```

Gambar 4.4: Hasil Running Program PCA

Dari gambar di atas, dapat kita lihat bahwa peleburan data atau reduksi dimensi menggunakan PCA pada data biner adalah teknik yang sangat efektif untuk mengurangi kompleksitas dan redundansi dalam dataset. Meskipun PCA pada data biner membutuhkan modifikasi tertentu, seperti penggunaan matriks korelasi, teknik ini dapat mengurangi jumlah fitur yang digunakan tanpa mengorbankan informasi penting yang terkandung dalam data. Penerapan PCA pada dataset biner, seperti dataset *Heart Failure Clinical Records*, dapat membantu meningkatkan efisiensi komputasi dan menghasilkan analisis *clustering* yang lebih akurat dan mudah diinterpretasikan. Penggunaan PCA pada data biner memungkinkan analisis *clustering* berjalan lebih cepat, lebih efisien, dan lebih dapat diandalkan, sekaligus mengurangi risiko *overfitting*. Namun, pemilihan jumlah komponen utama yang tepat harus dilakukan dengan hati-hati untuk menjaga keseimbangan antara kompleksitas dan informasi yang dihasilkan.

### 4.1.3 Standarisasi Data

Standarisasi data adalah salah satu langkah penting dalam proses *pre-processing* data yang bertujuan untuk memastikan data berada dalam skala yang seragam sebelum diterapkan dalam analisis lebih lanjut. Proses ini sangat krusial, terutama ketika dataset yang digunakan memiliki fitur dengan unit yang berbeda atau rentang nilai yang berbeda. Tanpa standarisasi, algoritma analisis data, seperti *clustering*, regresi, dan klasifikasi, dapat terpengaruh oleh perbedaan skala antar fitur, yang berpotensi menghasilkan model yang tidak optimal.

Standarisasi data merujuk pada teknik untuk mengubah fitur-fitur dalam dataset sehingga memiliki distribusi dengan rata-rata (*mean*) 0 dan deviasi standar (*standard*

*deviation*). Tujuan utama dari standarisasi adalah untuk menyeimbangkan kontribusi tiap fitur dalam model, menghindari dominasi fitur dengan rentang nilai yang lebih besar, dan meningkatkan performa algoritma analisis data yang sensitif terhadap skala data.

Sebelum melakukan standarisasi kita perlu menggabungkan data *non biner* dengan data yang sudah di PCA. Berikut kode penggabungan data *non-biner* dengan data yang sudah di PCA dengan menggunakan bantuan *platform* google colab berbasis python:

```
data = pd.concat([data_non_biner, pca_df], axis=1)
print(data)
```

	age	creatinine_phosphokinase	ejection_fraction	platelets	\
0	75.0	582	20	265000.00	
1	55.0	7861	38	263358.03	
2	65.0	146	20	162000.00	
3	50.0	111	20	210000.00	
4	65.0	160	20	327000.00	
..	...	...	...	...	...
294	62.0	61	38	155000.00	
295	55.0	1820	38	270000.00	
296	45.0	2060	60	742000.00	
297	45.0	2413	38	140000.00	
298	50.0	196	45	395000.00	

	serum_creatinine	serum_sodium	time	PCA
0	1.9	130	4	0.179355
1	1.1	136	6	0.361319
2	1.3	129	7	0.960308
3	1.9	137	7	0.100695
4	2.7	116	8	-0.903569
..	...	...	...	...
294	1.1	143	270	0.410196
295	1.2	139	271	-0.274796
296	0.8	138	278	-0.642945
297	1.4	140	280	0.960308
298	1.6	136	285	0.960308

Gambar 4.5: Penggabungan data *non-biner* dan data yang sudah di PCA

Setelah data digabungkan kita dapat melakukan standarisasi data dengan menggunakan bantuan *platform* google colab berbasis python:

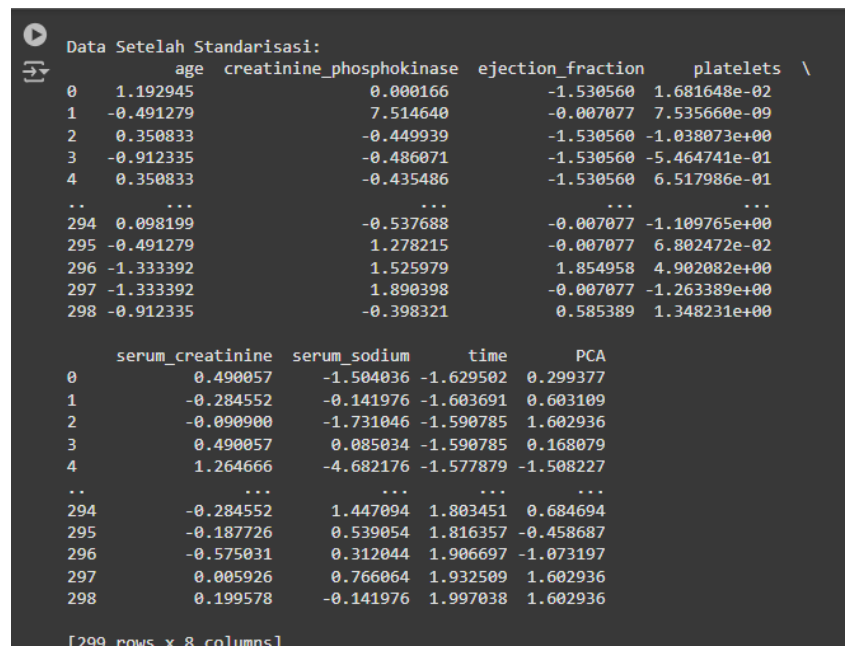
```
from sklearn.preprocessing import StandardScaler
import pandas as pd

# Inisialisasi StandardScaler
scaler = StandardScaler()

# Transformasi data
data_standar = scaler.fit_transform(data)

# Konversi hasil ke DataFrame
df_standar = pd.DataFrame(data_standar, columns=data.columns)
print("\nData Setelah Standarisasi:\n", df_standar)
# df_standar.to_csv('df_standar.csv', index=False)
```

Gambar 4.6: Program Standarisasi Data



Data Setelah Standarisasi:

	age	creatinine_phosphokinase	ejection_fraction	platelets
0	1.192945	0.000166	-1.530560	1.681648e-02
1	-0.491279	7.514640	-0.007077	7.535660e-09
2	0.350833	-0.449939	-1.530560	-1.038073e+00
3	-0.912335	-0.486071	-1.530560	-5.464741e-01
4	0.350833	-0.435486	-1.530560	6.517986e-01
...	...	...	...	...
294	0.098199	-0.537688	-0.007077	-1.109765e+00
295	-0.491279	1.278215	-0.007077	6.802472e-02
296	-1.333392	1.525979	1.854958	4.902082e+00
297	-1.333392	1.890398	-0.007077	-1.263389e+00
298	-0.912335	-0.398321	0.585389	1.348231e+00

	serum_creatinine	serum_sodium	time	PCA
0	0.490057	-1.504036	-1.629502	0.299377
1	-0.284552	-0.141976	-1.603691	0.603109
2	-0.090900	-1.731046	-1.590785	1.602936
3	0.490057	0.085034	-1.590785	0.168079
4	1.264666	-4.682176	-1.577879	-1.508227
...	...	...	...	...
294	-0.284552	1.447094	1.803451	0.684694
295	-0.187726	0.539054	1.816357	-0.458687
296	-0.575031	0.312044	1.906697	-1.073197
297	0.005926	0.766064	1.932509	1.602936
298	0.199578	-0.141976	1.997038	1.602936

[299 rows x 8 columns]

Gambar 4.7: Hasil running Program Standarisasi Data

Dari gambar di atas, dapat kita lihat bahwa standarisasi data memastikan bahwa setiap fitur, baik itu memiliki rentang nilai yang besar atau kecil, memiliki kontribusi yang setara dalam model analisis, dan dengan demikian, menghasilkan model yang lebih akurat dan mudah diinterpretasikan. Standarisasi data merupakan langkah krusial dalam mempersiapkan dataset untuk analisis lebih lanjut, terutama ketika menggunakan algoritma yang sensitif terhadap skala fitur. Proses ini mengubah data sehingga setiap fitur memiliki rata-rata 0 dan deviasi standar 1, yang memungkinkan analisis yang lebih adil dan efisien. Penerapan standarisasi sangat membantu dalam menghindari bias akibat perbedaan skala antar fitur dan meningkatkan performa algoritma analisis data seperti *clustering*.

## 4.2 Penerapan K-Means Clustering

Data mentah Heart Failure Clinical Record yang berasal dari UCIMLREPO setelah dilakukan pre-processing data, menghasilkan sebuah dataset yang bersih dan siap untuk dilanjutkan ke proses selanjutnya yaitu pembuatan Model menggunakan metode Clustering. Terdapat banyak algoritma yang dapat digunakan untuk metode Clustering tersebut, salah satunya dengan menggunakan algoritma K-Means Clustering untuk membuat model Clustering. Pada penelitian ini dalam membuat model Clustering dan menghitung algoritma K-Means, digunakan program komputer dengan bahasa pemrograman python yang cocok untuk menghitung data-data dan membuat model seperti Clustering.

### 4.2.1 Penerapan Algoritma Menggunakan Cara Manual

Sebelum menggunakan pemrograman untuk melakukan perhitungan K-Means Clustering, tentunya perlu paham terlebih dahulu tentang proses perhitungan K-Means

Clustering secara manual. Untuk mempermudah memahami proses perhitungan K-Means Clustering secara manual, maka digunakan 10 data pertama dari dataset yang sudah bersih.

Pasien	Age	Creatinine_Phosphokinase	Ejection_Fraction	Platelets	Serum_Creatinine	Serum_Sodium	Time	PCA
1	1.192945231	0.000165728	-1.530559528	0.016816484	0.490056987	-1.504036122	-1.629502414	0.299377408
2	-0.491279276	7.514639529	-0.00707675	7.54E-09	-0.284552352	-0.141976151	-1.603690737	0.603108844
3	0.350832977	-0.449938761	-1.530559528	-1.038073134	-0.090900017	-1.731046117	-1.590784898	1.602935854
4	-0.912335403	-0.486071002	-1.530559528	-0.546474088	0.490056987	0.085033844	-1.590784898	0.16807943
5	0.350832977	-0.435485864	-1.530559528	0.651798584	1.264666327	-4.682176055	-1.57787906	-1.508226664
6	2.45611361	-0.552141386	0.162199114	-0.607923969	0.683709322	-1.050016132	-1.57787906	0.864175003
7	1.192945231	-0.346703786	-1.953749189	-1.396530771	-0.187726185	0.085033844	-1.552067382	0.16807943
8	-0.070223149	-0.275471654	1.854957756	1.952487725	-0.284552352	-1.277026127	-1.552067382	0.553396328
9	0.350832977	-0.438582914	2.278147417	7.54E-09	0.102752318	0.31204384	-1.552067382	-0.458687138
10	1.614001357	-0.473682805	-0.260990546	1.276539038	7.752019547	-0.823006137	-1.552067382	0.864175003

Tabel 4.1: 10 Data Pasien

Untuk melakukan perhitungan menggunakan algoritma K-Means Clustering dengan data yang sudah bersih perlu dilakukan beberapa langkah, yaitu:

1. Menentukan Jumlah Cluster (K)

Jumlah Cluster merupakan tahap awal dan juga penentu dalam pengelompokan hasil Kalsterisasi. Dalam menentukan jumlah kluster yang sesuai, dapat memperhatikan beberapa aspek yang penting, salah satunya seperti kegunaan cluster dalam penelitian. Oleh karena itu, jumlah Cluster yang dipilih yaitu sebanyak 2, hal ini karena sesuai dengan kebutuhan Cluster untuk penelitian yaitu pengelompokan pasien gagal jantung yang memiliki risiko tinggi untuk meninggal atau tidak.

2. Menentukan Titik Pusat Cluster

Setelah menentukan jumlah kluster yang dibutuhkan, langkah selanjutnya yaitu menentukan titik pusat untuk masing-masing cluster atau yang disebut dengan titik Centroid. Dalam menentukan titik centroid peneliti dapat memilih secara acak posisi centroid namun dengan syarat centroid tersebut masih berada pada range nilai dari setiap atribut atau fitur yang dimiliki oleh data.

Dalam proses ini peneliti memilih titik centroid secara acak dengan mengambil salah satu nilai dari data untuk masing-masing centroid, yaitu peneliti data nomor urut ke-1 dan ke-8.

Cluster	Age	Creatinine_Phosphokinase	Ejection_Fraction	Platelets	Serum_Creatinine	Serum_Sodium	Time	PCA
Centroid 1	1.192945231	0.000165728	-1.530559528	0.016816484	0.490056987	-1.504036122	-1.629502414	0.299377408
Centroid 2	-0.070223149	-0.275471654	1.854957756	1.952487725	-0.284552352	-1.277026127	-1.552067382	0.553396328

Tabel 4.2: Data Centroid

3. Menghitung Jarak Data ke Setiap Centroid

Setelah memperoleh titik tengah dari setiap cluster, dilanjutkan dengan menghitung jarak dari data ke setiap titik tengah cluster atau titik centroid. Untuk menghitung jarak data ke titik centroid rumus untuk menghitungnya, yaitu rumus Euclidean Distance.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

**Keterangan:**



- $d(\mathbf{p}, \mathbf{q})$  : jarak antara titik  $\mathbf{p}$  and titik  $\mathbf{q}$ .
- $\mathbf{p} = (p_1, p_2, \dots, p_n)$  : titik centroid pada fitur ke- $n$ -dimensi.
- $\mathbf{q} = (q_1, q_2, \dots, q_n)$  : titik data pada fitur ke- $n$ -dimensi.
- $n$  : banyaknya fitur pada data

karena pada dataset terdapat 8 fitur sehingga dimensi  $n = 8$ , maka rumusnya yaitu:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2 + (p_4 - q_4)^2 + (p_5 - q_5)^2 + (p_6 - q_6)^2 + (p_7 - q_7)^2 + (p_8 - q_8)^2}$$

Dengan contoh, perhitungan jarak data pasien 2 ke centroid 1 yaitu:

- Data pasien 2 =  $(-0.491279276, 7.514639529, -0.00707675, 7.54E - 09, -0.284552352, -0.141976151, -1.603690737, 0.603108844)$
- titik centroid 1 =  $(1.192945231, 0.000165728, -1.530559528, 0.016816484, 0.490056987, -1.504036122, -1.629502414, 0.299377408)$

Maka nilai jaraknya ialah:

$$d = \sqrt{(1.192945231 - -0.491279276)^2 + (0.000165728 - 7.514639529)^2 + (-1.530559528 - -0.00707675)^2 + (0.016816484 - 7.54E - 09)^2 + (0.490056987 - -0.284552352)^2 + (-1.504036122 - -0.141976151)^2 + (-1.629502414 - -1.603690737)^2 + (0.299377408 - 0.603108844)^2}$$

$$d = \sqrt{(0.701665955)^2 + (7.514805257)^2 + (-1.537636278)^2 + (0.016816492)^2 + (0.205504635)^2 + (-1.646012273)^2 + (-3.233193151)^2 + (0.902486252)^2}$$

$$d = \sqrt{0.492335112 + 56.47229805 + 2.364325323 + 0.000282794 + 0.042232155 + 2.709356403 + 10.45353795 + 0.814481435}$$

$$d = \sqrt{73.34884923}$$

$$d = 8.010827503$$

Diperoleh jarak dari data pasien 2 ke centroid 1 sebesar 8.010827503, dengan menggunakan rumus dan cara yang sama, maka diperoleh nilai jarak dari data ke titik-titik centroidnya, yaitu:

#### 4. Menempatkan Data ke Dalam kluster

Setelah mendapatkan nilai jarak dari data ke titik-titik centroid, maka tahap selanjutnya yaitu mengelompokkan data ke dalam cluster yang sudah ada. Proses mengelompokkan data dapat dengan melihat jarak data ke titik centroid terkecil. Sebagai contoh pada pasien 1, 2 dan 3 dapat di kelompokkan ke kluster 1 karena jarak terkecilnya yaitu jarak dari titik data ke titik centroid 1, sedangkan pada pasien 8 dan 9 dapat dikelompokkan ke kluster 2 karena jarak terkecilnya yaitu dari titik data ke titik centroid 2. Dengan menggunakan cara yang sama, diperoleh hasil pengelompokan data ke kluster pada tabel berikut.

Pasien	Centroid 1	Centroid 2
1	0	4.195497076
2	8.010827503	8.332825214
3	2.02838543	4.686104588
4	2.744050024	4.589688193
5	3.908047868	5.621609421
6	2.39200055	4.119026251
7	2.303199328	5.416261961
8	4.195497076	0
9	4.408740657	2.809742431
10	7.558136118	8.526385466

Tabel 4.3: Jarak Data ke Centroid

Pasien	Centroid 1	Centroid 2	Cluster
1	0	4.195497076	klsuter 1
2	8.010827503	8.332825214	klsuter 1
3	2.02838543	4.686104588	klsuter 1
4	2.744050024	4.589688193	klsuter 1
5	3.908047868	5.621609421	klsuter 1
6	2.39200055	4.119026251	klsuter 1
7	2.303199328	5.416261961	klsuter 1
8	4.195497076	0	klsuter 2
9	4.408740657	2.809742431	klsuter 2
10	7.558136118	8.526385466	klsuter 1

Tabel 4.4: Pengelompokan Data

#### 5. Menentukan Titik Centroid Baru

Setelah terbentuknya cluster data dari proses sebelumnya, maka untuk selanjutnya mencari titik centroid baru dari kluster yang telah terbentuk dari data. Proses untuk menentukan titik centroid baru yaitu dengan menghitung rata-rata tiap fitur pada tiap kluster dengan data nilai awal. Rumus untuk mencari titik centroid baru dapat ditulis sebagai berikut.

$$\mu_n = \frac{1}{k} \sum_{i=1}^k x_i$$

Keterangan:

- $n$  : banyaknya dimensi atau fitur pada data.
- $k$  : banyaknya data yang berada pada kluster
- $\sum_{i=1}^k x_i$  : penjumlahan data ke  $i$  pada kluster dengan fitur  $n$  sampai data ke  $k$
- $\mu_n$  : nilai rata-rata pada fitur  $n$

Dalam menentukan titik centroid baru pada kluster, dilakukan perhitungan menggunakan rumus rata-rata dengan nilai  $n = 8$ , nilai  $k$  untuk kluster 1 = 8 dan nilai  $k$  untuk kluster 2 = 2. Proses dalam menentukan titik centroid pada Cluster 2, sebagai berikut.

### Cluster 2

$$\mu_1 = \frac{1}{2} \sum_{i=1}^2 x_i = \frac{1}{2} \times (-0.070223149 + 0.350832977) = 0.140304914$$

$$\mu_2 = \frac{1}{2} \sum_{i=1}^2 x_i = \frac{1}{2} \times (-0.275471654 + (-0.438582914)) = -0.357027284$$

$$\mu_3 = \frac{1}{2} \sum_{i=1}^2 x_i = \frac{1}{2} \times (1.854957756 + 2.278147417) = 2.066552587$$

$$\mu_4 = \frac{1}{2} \sum_{i=1}^2 x_i = \frac{1}{2} \times (1.952487725 + 7.54E - 09) = 0.976243866$$

$$\mu_5 = \frac{1}{2} \sum_{i=1}^2 x_i = \frac{1}{2} \times (-0.284552352 + 0.102752318) = -0.090900017$$

$$\mu_6 = \frac{1}{2} \sum_{i=1}^2 x_i = \frac{1}{2} \times (-1.277026127 + 0.31204384) = -0.482491144$$

$$\mu_7 = \frac{1}{2} \sum_{i=1}^2 x_i = \frac{1}{2} \times (-1.552067382 + (-1.552067382)) = -1.552067382$$

$$\mu_8 = \frac{1}{2} \sum_{i=1}^2 x_i = \frac{1}{2} \times (0.16807943 + 0.553396328) = 0.047354595$$

Maka diperoleh titik centroid pada cluster 2 = (0.140304914, -0.357027284, 2.066552587, 0.976243866, -0.090900017, -0.482491144, -1.552067382, 0.047354595). Dengan menggunakan rumus dan cara yang sama, titik centroid pada 2 cluster tersebut yaitu

Cluster	Age	Creatinine_Phosphokinase	Ejection_Fraction	Platelets	Serum_Creatinine	Serum_Sodium	Time	PCA
Centroid 1	0.719257088	0.596347707	-1.022731935	-0.205480981	1.264666327	-1.220273628	-1.584331979	0.382713039
Centroid 2	0.140304914	-0.357027284	2.066552587	0.976243866	-0.090900017	-0.482491144	-1.552067382	0.047354595

Tabel 4.5: Data Centroid Cluster Baru

### 6. Verifikasi Titik Centroid Baru

Setelah mendapatkan titik centroid baru, tahap selanjutnya yaitu memverifikasi titik centroid yang baru dengan titik centroid sebelumnya. Proses perhitungan K-Means Clustering ketika nilai titik centroid baru sama dengan nilai titik centroid sebelumnya, maka proses K-Means Clustering dikatakan selesai dan data telah ditempatkan pada kluster yang sesuai. Proses perhitungan K-Means Clustering

ketika nilai titik centroid baru berbeda dengan centroid yang sebelumnya, maka proses K-Means Clustering masih tetap dilanjutkan dengan iterasi mulai dari langkah ke-3 hingga langkah ke-6, iterasi tersebut masih tetap terus dilakukan hingga titik centroid yang baru sama dengan titik sentroid yang sebelumnya.

## 4.2.2 Penerapan Algoritma Menggunakan Platform Google Colab

Setelah menggunakan cara manual, berikut akan dijelaskan algoritma K-Means menggunakan bantuan platform google colab

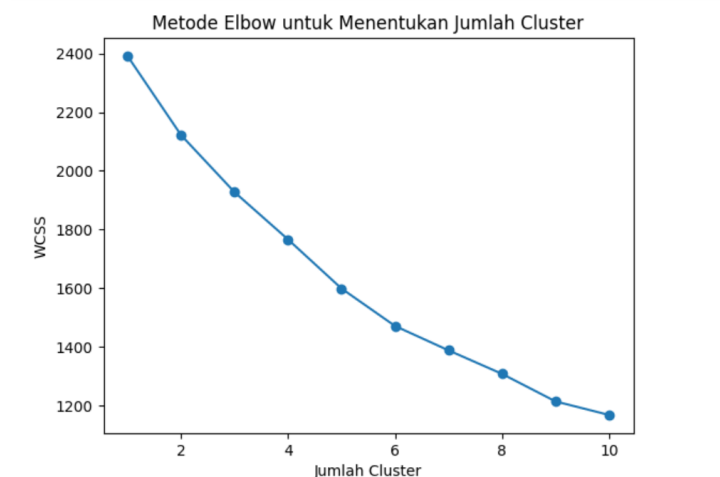
### 1. Membuat model klustering menggunakan metode K-Means

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

wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=4, random_state=92)
    kmeans.fit(df_standar)
    wcss.append(kmeans.inertia_)

plt.plot(range(1, 11), wcss, marker='o')
plt.title('Metode Elbow untuk Menentukan Jumlah Cluster')
plt.xlabel('Jumlah Cluster')
plt.ylabel('WCSS')
plt.show()
```

Gambar 4.8: Program Membuat Model Klustering Menggunakan Metode K-Means



Gambar 4.9: Grafik Ellbow

Diperoleh nilai K optimal 2

```
[ ] from sklearn.cluster import KMeans

optimal_k = 2

kmeans = KMeans(n_clusters=optimal_k, init='k-means++', max_iter=1000, n_init=4, random_state=92)
y_kmeans = kmeans.fit_predict(df_standar)
kmeans.inertia_

2122.3414277218867
```

Gambar 4.10: Hasil Running Program

Pada proses ini, dibuat model clustering menggunakan metode K-Means dengan tujuan untuk mengelompokkan data pasien gagal jantung berdasarkan karakteristik tertentu. Langkah awal dalam menggunakan K-Means adalah menentukan jumlah cluster (KKK) yang optimal. Penentuan nilai KKK dilakukan menggunakan metode Elbow, yang merupakan salah satu pendekatan visual untuk menilai jumlah cluster terbaik berdasarkan variasi dalam data. Pada penelitian ini kami menggunakan python untuk menghitung jumlah kluster yang optimal menggunakan metode elbow.

Grafik Elbow yang dihasilkan menunjukkan penurunan nilai Within-Cluster Sum of Squares (WCSS) seiring bertambahnya jumlah cluster. Pada titik tertentu, penurunan WCSS mulai melambat secara signifikan, menciptakan "titik siku" (elbow point), yang menjadi dasar untuk memilih jumlah cluster optimal. Dalam kasus ini, K=2 dianggap sebagai pilihan terbaik karena memberikan keseimbangan antara meminimalkan WCSS dan menghindari kompleksitas model yang berlebihan. Setelah menentukan jumlah cluster optimal, algoritma K-Means dilatih ulang dengan K=2 menggunakan parameter seperti iterasi maksimum sebanyak 1000 untuk memastikan model mencapai konvergensi. Hasil clustering berupa label prediksi yang mengelompokkan setiap data ke salah satu dari dua cluster. Nilai WCSS terakhir menunjukkan seberapa baik data dikelompokkan dalam dua cluster tersebut, dengan nilai yang lebih kecil menunjukkan bahwa data dalam cluster memiliki jarak yang lebih kecil terhadap centroid masing-masing.

## 2. Visualisasi hasil model dengan kluster dan centroid kluster

```
list_feature = ['age', 'creatinine_phosphokinase', 'ejection_fraction', 'platelets', 'serum_creatinine', 'serum_sodium', 'time', 'PCA']
for k in list_feature:
    for m in list_feature:
        if k != m:
            plt.scatter(df_standar[k], df_standar[m], c=kmeans.labels_)
            plt.scatter(kmeans.cluster_centers_[:,2,list_feature.index(k)], kmeans.cluster_centers_[:,2,list_feature.index(m)],
                        s=300, c='red', markers='.', label='Centroid')
            for i, center in enumerate(kmeans.cluster_centers_):
                plt.text(center[list_feature.index(k)], center[list_feature.index(m)], f'K{i+1}', color='black', fontsize=12, ha='center', va='center')
            plt.title("Visualisasi Cluster")
            plt.xlabel(k)
            plt.ylabel(m)
            plt.legend()
            plt.show()
```

Gambar 4.11: Program Visualisasi Hasil Model

```
from sklearn.model_selection import RandomizedSearchCV

def find_best_kmeans(data, n_clusters, n_iterations=100):
    param_dist = {'n_init': range(100, 151), 'random_state': range(100)}

    kmeans = KMeans(n_clusters=n_clusters, init='k-means++', max_iter=1000)

    random_search = RandomizedSearchCV(kmeans, param_distributions=param_dist, n_iter=n_iterations, cv=5, random_state=42)

    random_search.fit(data)

    return random_search.best_estimator_, random_search.best_score_

a, b = find_best_kmeans(df_standar, 2)

print(a)
print(b)
```

KMeans(max\_iter=1000, n\_clusters=2, n\_init=137, random\_state=4)  
-497.3346835229394

Gambar 4.12: Program Mengecek Hasil Akurasi Model

```

# while True:

for j in range(100):
    kmeans = KMeans(n_clusters=optimal_k, init='k-means++', max_iter=1000, n_init=4, random_state=j)
    y_kmeans = kmeans.fit_predict(df_standar)
    kmeans.inertia_

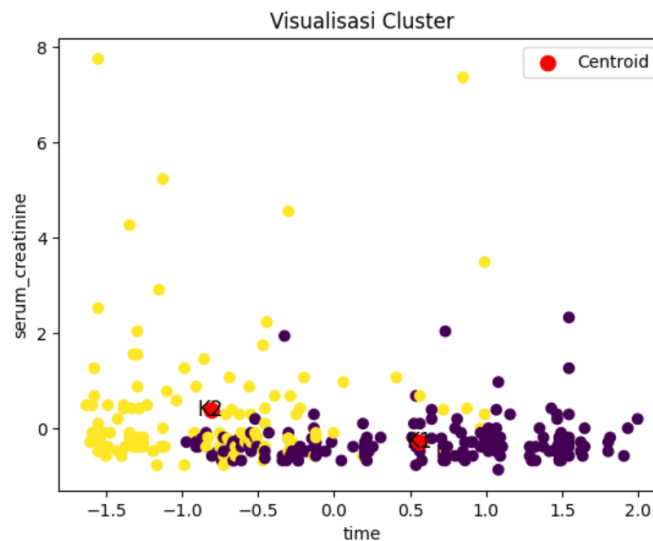
    y = pd.DataFrame(y)
    y_list = []
    for index, i in y.iterrows():
        y_list.append(i[0])
    correct_labels = sum(y_list == y_kmeans)

    # print(f"nilai nilai_random_state : {j}")
    # print("Result: %d out of %d samples were correctly labeled." % (correct_labels, len(y_list)))
    # print('Accuracy score: {0:0.2f}'.format(correct_labels/float(len(y))))
    if correct_labels/float(len(y)) > nilai_akurasi:
        nilai_akurasi = correct_labels/float(len(y))
        nilai_random_state = j

print(f"===== nilai akurasi {nilai_akurasi} dengan nilai_random_state : {nilai_random_state} =====")
print('Accuracy score: {0:0.2f}'.format(nilai_akurasi))
# print(f"n_init : {n_init}")

```

Gambar 4.13: Program Mengecek Hasil Akurasi Model



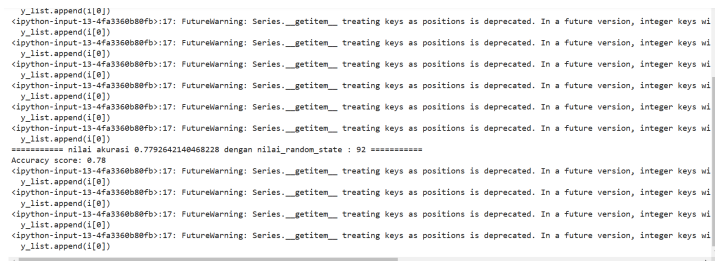
Gambar 4.14: Hasil Visualisasi Model

Hasil visualisasi cluster yang ditampilkan menunjukkan hubungan penting antara waktu penanganan (time) dan kadar kreatinin serum (serum creatinine) dalam kaitannya dengan prognosis pasien gagal jantung, yang secara langsung berkaitan dengan kehidupan dan kematian seseorang. Serum creatinine merupakan indikator fungsi ginjal, di mana kadar yang tinggi menandakan adanya gangguan ginjal yang sering kali berhubungan dengan kondisi gagal jantung yang lebih parah. Sementara itu, variabel time mencerminkan durasi perawatan pasien atau seberapa cepat mereka menerima intervensi medis.

Hubungan antara kedua variabel ini dapat dilihat dari pola distribusi data dalam kluster. Pasien dengan kadar kreatinin serum yang lebih tinggi (kluster kuning) cenderung memiliki waktu perawatan yang lebih bervariasi, yang mengindikasikan bahwa beberapa pasien mungkin sudah dalam kondisi kritis sebelum mendapatkan penanganan yang memadai. Sebaliknya, pasien dengan kadar kreatinin serum yang lebih rendah (kluster ungu) menunjukkan pola yang lebih konsisten dalam waktu penanganan, yang mungkin mencerminkan efektivitas intervensi medis pada tahap yang lebih awal.

Korelasi ini menunjukkan bahwa waktu penanganan yang lebih cepat dapat membantu mengurangi dampak buruk dari kadar kreatinin serum yang tinggi, sehingga meningkatkan peluang hidup pasien. Sebaliknya, jika penanganan terlambat, kadar kreatinin yang tinggi dapat memperburuk kondisi pasien, mempercepat kerusakan organ, dan meningkatkan risiko kematian. Dengan memahami hubungan antara kedua variabel ini, tenaga medis dapat memanfaatkan kluster untuk mengidentifikasi pasien yang membutuhkan penanganan segera, sehingga dapat menyelamatkan lebih banyak nyawa dan meningkatkan kualitas hidup pasien secara keseluruhan.

### 3. Mengecek hasil akurasi model



```

y_list.append(i[0])
<ipython-input-13-4fa3360b08fb>:17: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
y_list.append(i[1])
<ipython-input-13-4fa3360b08fb>:17: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
y_list.append(i[0])
<ipython-input-13-4fa3360b08fb>:17: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
y_list.append(i[1])
<ipython-input-13-4fa3360b08fb>:17: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
y_list.append(i[0])
<ipython-input-13-4fa3360b08fb>:17: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
y_list.append(i[1])
<ipython-input-13-4fa3360b08fb>:17: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
y_list.append(i[0])
<ipython-input-13-4fa3360b08fb>:17: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
y_list.append(i[1])
Accuracy score: 0.78
===== nilai akurasi 0.7792642140468228 dengan nilai_random_state : 92 =====
<ipython-input-13-4fa3360b08fb>:17: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
y_list.append(i[0])
<ipython-input-13-4fa3360b08fb>:17: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
y_list.append(i[1])
<ipython-input-13-4fa3360b08fb>:17: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
y_list.append(i[0])
<ipython-input-13-4fa3360b08fb>:17: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
y_list.append(i[1])

```

Gambar 4.15: Hasil Pengecekan Akurasi Model

Hasil dari kode yang sudah dijalankan menunjukkan bahwa proses pengujian dilakukan untuk menemukan nilai random state terbaik yang menghasilkan akurasi pengelompokan tertinggi dalam model K-Means. Setelah melalui iterasi sebanyak 100 kali, ditemukan bahwa nilai random state sebesar 92 menghasilkan akurasi tertinggi, yaitu 77,93 persen. Hal ini berarti, pada konfigurasi tersebut, model K-Means mampu mengelompokkan data dengan tingkat kesesuaian yang lebih baik dibandingkan nilai random state lainnya. Hasil ini mencerminkan pengaruh signifikan dari inisialisasi centroid terhadap performa pengelompokan.

Proses ini penting dilakukan karena nilai random state dalam algoritma K-Means memengaruhi posisi awal centroid, yang menentukan jalannya proses pengelompokan. Posisi awal centroid yang kurang optimal dapat menyebabkan model jatuh ke solusi lokal yang kurang baik, sehingga hasil *clustering* menjadi kurang representatif terhadap pola data sebenarnya. Dengan menguji berbagai nilai random state, dapat dipastikan bahwa model bekerja pada konfigurasi yang menghasilkan pengelompokan terbaik. Selain itu, mencari nilai random state terbaik membantu meningkatkan konsistensi hasil clustering. Karena K-Means adalah algoritma non-deterministik yang hasilnya dapat berbeda-beda tergantung pada inisialisasi, menemukan konfigurasi yang optimal memastikan bahwa pengelompokan yang dihasilkan memiliki akurasi tertinggi. Hal ini menjadi sangat penting terutama dalam penelitian berbasis kesehatan, seperti pada dataset gagal jantung, di mana kualitas pengelompokan dapat memengaruhi interpretasi pola data dan pengambilan keputusan klinis.

## Bab 5

# Kesimpulan dan Saran

### 5.1 Kesimpulan

Berdasarkan hasil penelitian dan analisis data yang telah dilakukan, dapat disimpulkan bahwa penerapan algoritma K-Means *clustering* pada dataset pasien gagal jantung berhasil mengelompokkan pasien berdasarkan karakteristik klinis dan laboratorium, seperti usia, jenis kelamin, anemia, kadar enzim kreatinin fosfokinase, penderita diabetes, fraksi ejeksi, penderita hipertensi, trombosit, kadar kreatinin serum, kadar natrium serum, riwayat merokok dan waktu penanganan. Hasil pengelompokan menunjukkan adanya beberapa pola atau kelompok utama yang dapat diidentifikasi, 2 kelompok yaitu pasien yang memiliki risiko kematian tinggi dan pasien yang memiliki risiko kematian rendah. Variabel-variabel seperti usia, jenis kelamin, anemia, kadar enzim kreatinin fosfokinase, penderita diabetes, fraksi ejeksi, penderita hipertensi, trombosit, kadar kreatinin serum, kadar natrium serum, riwayat merokok dan waktu penanganan terbukti memiliki kontribusi signifikan dalam menentukan pengelompokan pasien, dengan ejection fraction dan usia menjadi faktor dominan dalam memisahkan kelompok berisiko tinggi. Hasil 2 pengelompokan ini dapat membantu pengambilan keputusan klinis dalam pengelolaan pasien gagal jantung, dengan memberikan gambaran yang lebih jelas tentang kelompok pasien yang membutuhkan perhatian lebih, pemantauan ketat, atau terapi yang lebih agresif. Dengan demikian, penerapan K-Means dalam penelitian ini terbukti efektif untuk meningkatkan pemahaman tentang faktor risiko dan prediksi mengenai perkembangan penyakit pada pasien gagal jantung.

### 5.2 Saran

Berdasarkan hasil penelitian ini, beberapa saran yang dapat diajukan untuk pengembangan lebih lanjut dan implementasi praktis, sebagai berikut:

1. Pada penelitian selanjutnya disarankan untuk mengumpulkan data yang digunakan lebih lengkap dengan mencakup lebih banyak variabel klinis dan biokimiawi yang relevan, seperti faktor gaya hidup dan kondisi komorbiditas lainnya, sehingga nilai akurasi dan hasil pengelompokan lebih meningkat.
2. Dengan menggunakan algoritma K-Means dapat berhasil mengidentifikasi kelompok pasien dengan baik, namun diharapkan untuk melakukan klustering



dengan algoritma yang lain juga seperti DBSCAN atau hierarchical *clustering* untuk mendapatkan pengelompokan pasien yang optimal.

3. Hasil pengelompokan sebaiknya dikombinasikan dengan model pembelajaran mesin lainnya, seperti pohon keputusan atau regresi logistik, untuk membangun model prediktif yang dapat memprediksi risiko kematian pada pasien gagal jantung.
4. Dalam mengimplementasi praktis hasil dari penelitian, perhatikan pada aspek-aspek berikut:
  - Rumah sakit dan fasilitas kesehatan dapat mengadakan pelatihan untuk tenaga medis mengenai interpretasi hasil pengelompokan pasien dan bagaimana hasil tersebut dapat digunakan dalam pengambilan keputusan klinis.
  - Mengembangkan sistem berbasis data yang dapat diintegrasikan ke dalam sistem informasi rumah sakit untuk otomatisasi pengelompokan pasien dan penyediaan rekomendasi perawatan sesuai dengan kelompok risiko.
  - Membuat protokol perawatan khusus untuk kelompok risiko tinggi, termasuk jadwal pemantauan yang lebih sering, pemeriksaan laboratorium tambahan, dan terapi yang lebih agresif.
5. Dengan menggunakan dataset yang lebih besar dan beragam, hasil *clustering* dapat digeneralisasi ke populasi yang lebih luas dan dapat diterapkan di berbagai bidang klinis.

# Bibliografi

- [1] Ali, M. M., Al-Doori, V. S., Mirzah, N., Hemu, A. A., Mahmud, I., Azam, S., Altabatabaie, K. F., Ahmed, K., Bui, F. M., & Moni, M. A. (2023). *A machine learning approach for risk factors analysis and survival prediction of Heart Failure patients. Healthcare Analytics*, 3. <https://doi.org/10.1016/j.health.2023.100182>
- [2] Chávez-Íñiguez, J. S., Ivey-Miranda, J. B., De la Vega-Mendez, F. M., & Borges-Vela, J. A. (2023). *How to interpret serum creatinine increases during decongestion*. In *Frontiers in Cardiovascular Medicine* (Vol. 9). Frontiers Media S.A. <https://doi.org/10.3389/fcvm.2022.1098553>
- [3] Chicco, D., & Jurman, G. (2020). *Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone*. *BMC Medical Informatics and Decision Making*, 20(1). <https://doi.org/10.1186/s12911-020-1023-5>
- [4] Dahlen, B., Schulz, A., Göbel, S., Tröbs, S. O., Schwuchow-Thonke, S., Spronk, H. M., Prochaska, J. H., Arnold, N., Lackner, K. J., Gori, T., ten Cate, H., Münzel, T., Wild, P. S., & Panova-Noeva, M. (2021). *The impact of platelet indices on clinical outcome in heart failure: results from the MyoVasc study*. *ESC Heart Failure*, 8(4), 2991–3001. <https://doi.org/10.1002/ehf2.13390>
- [5] Doifode, M. G., Aglave, M. H., Jaybhaye, D. S., Rawat, S., & Pawar, M. (2024). *EPRA International Journal of Research and Development (IJRD) CASE STUDY OF HEART FAILURE*. <https://doi.org/10.36713/epra2016>
- [6] Elendu, C., Amaechi, D. C., Elendu, T. C., Ashna, M., Ross-Comptis, J., Ansong, S. O., Egbunu, E. O., Okafor, G. C., Jingwa, K. A., Akintunde, A. A., Ogah, C. M., Edeko, M. O., Ibitoye, A. V., Ogunseye, M. O., Alakwe-Ojimba, C. E., Omeludike, E. K., Oguine, C. A., Afuh, R. N., Olawuni, C. A., ... Aborisade, O. (2023). *Heart failure and diabetes: Understanding the bidirectional relationship*. In *Medicine (United States)* (Vol. 102, Issue 37, p. E34906). Lippincott Williams and Wilkins. <https://doi.org/10.1097/MD.00000000000034906>
- [7] Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. [www.aaai.org](http://www.aaai.org)
- [8] Grote Beverborg, N., van Veldhuisen, D. J., & van der Meer, P. (2018). *Anemia in Heart Failure: Still Relevant?* In *JACC: Heart Failure* (Vol. 6, Issue 3, pp. 201–208). Elsevier Inc. <https://doi.org/10.1016/j.jchf.2017.08.023>
- [9] Halvorsen, S., Mehilli, J., Cassese, S., Hall, T. S., Abdelhamid, M., Barbato, E., De Hert, S., De Laval, I., Geisler, T., Hinterbuchner, L., Ibanez, B., Lenarczyk, R.,

- Mansmann, U. R., McGreavy, P., Mueller, C., Muneretto, C., Niessner, A., Potpara, T. S., Ristić, A., ... Zacharowski, K. (2022). 2022 ESC Guidelines on cardiovascular assessment and management of patients undergoing non-cardiac surgery. In *European Heart Journal* (Vol. 43, Issue 39, pp. 3826–3924). Oxford University Press. <https://doi.org/10.1093/eurheartj/ehac270>
- [10] Heidenreich, P. A., Bozkurt, B., Aguilar, D., Allen, L. A., Byun, J. J., Colvin, M. M., Deswal, A., Drazner, M. H., Dunlay, S. M., Evers, L. R., Fang, J. C., Fedson, S. E., Fonarow, G. C., Hayek, S. S., Hernandez, A. F., Khazanie, P., Kittleson, M. M., Lee, C. S., Link, M. S., ... Yancy, C. W. (2022). 2022 AHA/ACC/HFSA Guideline for the Management of Heart Failure: A Report of the American College of Cardiology/American Heart Association Joint Committee on Clinical Practice Guidelines. *Journal of the American College of Cardiology*, 79(17), e263–e421. <https://doi.org/10.1016/j.jacc.2021.12.012>
- [11] Jain, A. K., Murty, M. N., & Flynn, P. J. (2000). *Data Clustering: A Review*.
- [12] Khanna, D., Sahu, R., Baths, V., & Deshpande, B. (2015). *Comparative Study of Classification Techniques (SVM, Logistic Regression and Neural Networks) to Predict the Prevalence of Heart Disease*. *International Journal of Machine Learning and Computing*, 5(5), 414–419. <https://doi.org/10.7763/ijmlc.2015.v5.544>
- [13] Savarese, G., Becher, P. M., Lund, L. H., Seferovic, P., Rosano, G. M. C., & Coats, A. J. S. (2022). *Global burden of heart failure: a comprehensive and updated review of epidemiology*. In *Cardiovascular Research* (Vol. 118, Issue 17, pp. 3272–3287). Oxford University Press. <https://doi.org/10.1093/cvr/cvac013>
- [14] Zhang, L., Huang, T., Xu, F., Li, S., Zheng, S., Lyu, J., & Yin, H. (2022). *Prediction of prognosis in elderly patients with sepsis based on machine learning (random survival forest)*. *BMC Emergency Medicine*, 22(1). <https://doi.org/10.1186/s12873-022-00582-z>

# Lampiran A

## Source Code

Tuliskan *source code* disini. Berikut adalah contoh *source code*:

```
1
2 # Menginstall library ucimlrepo
3 # pip install ucimlrepo
4
5
6
7 # Mendownload data
8 from ucimlrepo import fetch_ucirepo
9
10 # fetch dataset
11 heart_failure_clinical_records = fetch_ucirepo(id=519)
12
13 # data (as pandas dataframes)
14 x = heart_failure_clinical_records.data.features
15 y = heart_failure_clinical_records.data.targets
16
17 print(x)
18 print(y)
19
20
21
22 # Mengecek data kosong
23 x.isnull().sum()
24
25
26
27 # Membagi data dan melakukan PCA
28 import numpy as np
29 import pandas as pd
30 from sklearn.decomposition import PCA
31 from sklearn.preprocessing import StandardScaler
32
33 # print(X)
34 feature_pca = [
```

```

35     'anaemia',
36     'diabetes',
37     'high_blood_pressure',
38     'sex',
39     'smoking']
40 feature = []
41
42 for i in x:
43     if i not in feature_pca:
44         feature.append(i)
45
46 data_biner = x[feature_pca]
47 data_non_biner = x[feature]
48
49 print(data_biner)
50 print(data_non_biner)
51
52 # proses PCA
53 pca = PCA(n_components=1)
54 pca_data = pca.fit_transform(data_biner)
55
56 pca_df = pd.DataFrame(pca_data, columns=["PCA"])
57
58 print("Data_PCA:")
59 print(pca_df)
60
61
62
63 # Menggabungkan data dengan data PCA
64 data = pd.concat([data_non_biner, pca_df], axis=1)
65 print(data)
66
67
68
69 # Melakukan standarisasi data
70 from sklearn.preprocessing import StandardScaler
71 import pandas as pd
72
73 # Inisialisasi StandardScaler
74 scaler = StandardScaler()
75
76 # Transformasi data
77 data_standar = scaler.fit_transform(data)
78
79 # Konversi hasil ke DataFrame
80 df_standar = pd.DataFrame(data_standar, columns=data.columns)
81 print("\nData_Setelah_Standarisasi:\n", df_standar)
82

```

```
83
84
85 # Mencari nilai K dengan Ellbow
86 from sklearn.cluster import KMeans
87 import matplotlib.pyplot as plt
88
89 wcss = []
90 for i in range(1, 11):
91     kmeans = KMeans(
92         n_clusters=i,
93         init='k-means++',
94         max_iter=300,
95         n_init=4,
96         random_state=92)
97     kmeans.fit(df_standar)
98     wcss.append(kmeans.inertia_)
99
100 plt.plot(range(1, 11), wcss, marker='o')
101 plt.title('Metode_Elbow_untuk_Menentukan_Jumlah_Cluster')
102 plt.xlabel('Jumlah_Cluster')
103 plt.ylabel('WCSS')
104 plt.show()
105
106
107
108 # Membuat model klastering
109 from sklearn.cluster import KMeans
110
111 optimal_k = 2
112
113 kmeans = KMeans(
114     n_clusters=optimal_k,
115     init='k-means++',
116     max_iter=1000,
117     n_init=4,
118     random_state=92)
119 y_kmeans = kmeans.fit_predict(df_standar)
120 kmeans.inertia_
121
122
123
124 # Visualisasi data
125 plt.scatter(
126     df_standar['age'],
127     df_standar['ejection_fraction'],
128     c=kmeans.labels_)
129 plt.scatter(
130     kmeans.cluster_centers_[ :2,0],
```

```

131     kmeans.cluster_centers_[ :2,2],
132     s=300,
133     c='red',
134     marker='.',
135     label='Centroid')
136 plt.title("Visualisasi_Cluster")
137 plt.xlabel("age")
138 plt.ylabel("ejection_fraction")
139 plt.legend()
140 plt.show()
141
142
143
144 # Mengecek akurasi model
145 from sklearn.model_selection import RandomizedSearchCV
146
147 def find_best_kmeans(data, n_clusters, n_iterations=100):
148     param_dist = {
149         'n_init':
150             range(100, 151), 'random_state':
151             range(100)}
152
153     kmeans = KMeans(
154         n_clusters=n_clusters,
155         init='k-means++',
156         max_iter=1000)
157
158     random_search = RandomizedSearchCV(
159         kmeans,
160         param_distributions=param_dist,
161         n_iter=n_iterations,
162         cv=5,
163         random_state=42)
164
165     random_search.fit(data)
166
167     return random_search.best_estimator_, random_search.
168         ↪ best_score_
169
170
171 a, b = find_best_kmeans(df_standar, 2)
172
173 print(a)
174 print(b)

```