

BAB IV

HASIL DAN PEMBAHASAN

Penelitian ini menggunakan jumlah data sebanyak 21.241 dataset yang diperoleh di UPT Puskesmas Sumur Batu dengan 19 atribut yaitu NIK, Nama peserta deteksi dini, Kecamatan, Usia, JK, Status tekanan darah, Hasil tekanan darah (Sistol), Hasil tekanan darah (Diastol), Status gula darah sewaktu, Hasil gula darah sewaktu, Status lingkaran perut, Hasil lingkaran perut, Tajam penglihatan, Tajam pendengaran, Risiko katarak, Status puma, Skor puma. Dalam penelitian ini menggunakan tools *Google Collab*, *Software Excel* dan *Tableau*.

4.1 Seleksi Fitur (*Feature Selection*)

Tahap ini, merupakan proses pemilihan atribut yang paling relevan dan signifikan dalam pemrosesan clustering. Adapun tahapannya yaitu :

1. Import Library

Import library berfungsi untuk memanfaatkan modul atau Pustaka yang berisi fungsi, kelas, dan metode yang telah tersedia, proses tersebut dapat dilihat pada gambar 4.1 berikut:

```
import numpy as np
import pandas as pd
import scipy

import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
from sklearn.preprocessing import StandardScaler
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import KMeans

import random
random.seed(15)
```

Gambar 4.1 *Import Library*

2. Import Data

Import data berfungsi untuk memuat dan membaca dataset ke dalam Google Collab agar dapat dianalisis dan diolah. Proses import data dapat dilihat pada gambar 4.2 berikut:

```
df_segmentation = pd.read_csv('/content/data_hipertensi.csv')
df_segmentation.head()
```

Gambar 4. 2 Import Data

Berikut dataset hasil dari import data yang dapat dilihat pada tabel 4.1:

Tabel 4. 1 Dataset

	NIK	Nama Peserta Deteksi Dem	Kecamatan	Usia	JK	Status - Tekanan Darah	Hasil - Tekanan Darah (Sistol)	Hasil - Tekanan Darah (Diastol)	Status - Gula Darah Sewaktu	Hasil - Gula Darah Sewaktu	Status - IMT	Hasil - IMT	Status - Lingkar Perut	Hasil - Lingkar Perut	Tajam Penglihatan	Tajam Pendengaran	Risiko Katarak	Status - FUMGA	Skor FUMGA
1	1871095501720000	NIRMALA	TELUKBETUNG UTARA	52	female	HIPERTENSI	110.0	90	HIPERGLIKEMIK	325	NORMAL	25.3	NORMAL	60	GANGGUAN BERAT	NORMAL	NORMAL	RISIKO RENDAH PPOK	5
2	1871095304720000	TITA TRIASTUTI	TELUKBETUNG UTARA	50	female	HIPERTENSI	140.0	90	HIPERGLIKEMIK	230	NORMAL	26.9	NORMAL	70	GANGGUAN BERAT	NORMAL	NORMAL	RISIKO RENDAH PPOK	5
3	1871091901880000	CAHYO FRARIAN	TELUKBETUNG UTARA	47	male	HIPERTENSI	140.0	90	HIPERGLIKEMIK	250	NORMAL	25.3	NORMAL	90	GANGGUAN BERAT	NORMAL	NORMAL	RISIKO RENDAH PPOK	5
4	1871094106720000	RAHAYU WAHYUNI	TELUKBETUNG UTARA	47	female	HIPERTENSI	140.0	90	HIPERGLIKEMIK	209	NORMAL	25.3	OBESTITAS SENTRAL	92	GANGGUAN BERAT	NORMAL	NORMAL	RISIKO RENDAH PPOK	6
5	1871090701720000	AHMAD ASRORI	TELUKBETUNG UTARA	47	male	HIPERTENSI	140.0	80	HIPERGLIKEMIK	250	BB LEBIH	25.6	OBESTITAS SENTRAL	88	NORMAL	NORMAL	NORMAL	RISIKO RENDAH PPOK	6
6	1871095501720000	SULIYAH	TELUKBETUNG UTARA	56	female	HIPERTENSI	140.0	80	HIPERGLIKEMIK	258	OBESTITAS	26.7	OBESTITAS SENTRAL	92	NORMAL	NORMAL	NORMAL	RISIKO RENDAH PPOK	7

3. Seleksi Fitur (*Feature Selection*)

Pada tahap ini atribut yang akan digunakan terdapat 10 atribut yaitu Usia, JK, Hasil tekanan darah (Sistol), Hasil tekanan darah (Diastol), Hasil gula darah sewaktu, Hasil lingkar perut, Tajam penglihatan, Tajam pendengaran, Risiko katarak, Skor puma. Pada atribut tersebutlah yang akan menjadi indikator dalam proses clustering. Proses tersebut dapat dilihat pada gambar 4.3 berikut:

```

import pandas as pd

# Memuat dataset dari path yang diberikan
dataset_path = '/content/data_hipertensi.csv'
data = pd.read_csv(dataset_path)

# Menentukan fitur yang ingin diseleksi
selected_features = [
    'Usia', 'JK', 'Hasil - Tekanan Darah (Sistol)',
    'Hasil - Tekanan Darah (Diastol)',
    'Hasil - Gula Darah Sewaktu', 'Hasil - IMT',
    'Hasil - Lingkar Perut', 'Tajam Penglihatan',
    'Tajam Pendengaran', 'Risiko Katarak', 'Skor PUMA']

# Melakukan seleksi fitur
selected_data = data[selected_features]

# Menampilkan informasi tentang data yang diseleksi
print("\nInformasi tentang data yang diseleksi:")
print(selected_data.info())

# Simpan DataFrame yang sudah dimodifikasi ke dalam file CSV
selected_data.to_csv('dataset_selection.csv', index=False)

```

Gambar 4.3 Proses Seleksi Fitur

Hasil seleksi fitur dapat dilihat pada gambar 4.4 berikut:

```

Informasi tentang data yang diseleksi:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21241 entries, 0 to 21240
Data columns (total 11 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Usia                                   21236 non-null  float64
1   JK                                     21241 non-null  object
2   Hasil - Tekanan Darah (Sistol)        21241 non-null  int64
3   Hasil - Tekanan Darah (Diastol)       21241 non-null  object
4   Hasil - Gula Darah Sewaktu            20015 non-null  float64
5   Hasil - IMT                           21241 non-null  float64
6   Hasil - Lingkar Perut                 19863 non-null  float64
7   Tajam Penglihatan                     19495 non-null  object
8   Tajam Pendengaran                     21226 non-null  object
9   Risiko Katarak                        21240 non-null  object
10  Skor PUMA                              20076 non-null  float64
dtypes: float64(5), int64(1), object(5)
memory usage: 1.8+ MB
None

```

Gambar 4.4 Hasil Seleksi Fitur

4.2 Cleaning Data

Cleaning data bertujuan untuk memastikan kualitas dataset yang akan digunakan. Proses ini meliputi identifikasi dan perbaikan kesalahan dalam data, seperti menghapus atau memperbaiki nilai yang hilang. Proses tersebut dapat dilihat pada gambar 4.5 sebagai berikut:

```
import pandas as pd

# Muat dataset
dataset_path = '/content/dataset_selection.csv'
df = pd.read_csv(dataset_path)

# Menampilkan jumlah missing values sebelum pembersihan
print("Jumlah missing values sebelum pembersihan:")
print(df.isnull().sum())

# Menghapus baris yang memiliki missing values
df_cleaned = df.dropna()

# Menampilkan jumlah missing values setelah pembersihan
print("\nJumlah missing values setelah pembersihan:")
print(df_cleaned.isnull().sum())

# Simpan DataFrame yang sudah dimodifikasi ke dalam file CSV
df_cleaned.to_csv('dataset_cleaning.csv', index=False)
```

Gambar 4.5 Proses Cleaning data

Hasil *cleaning data* tersisa 17.316 dataset, hasil tersebut dapat dilihat pada gambar 4.5 berikut:

```
Jumlah missing values sebelum pembersihan:
Usia          5
JK            0
Hasil - Tekanan Darah (Sistol)  0
Hasil - Tekanan Darah (Diastol)  0
Hasil - Gula Darah Sewaktu      1226
Hasil - IMT                     0
Hasil - Lingkar Perut           1378
Tajam Penglihatan              1746
Tajam Pendengaran              15
Risiko Katarak                 1
Skor PUMA                      1165
dtype: int64
```

```

Jumlah missing values setelah pembersihan:
Usia                0
JK                  0
Hasil - Tekanan Darah (Sistol)  0
Hasil - Tekanan Darah (Diastol)  0
Hasil - Gula Darah Sewaktu      0
Hasil - IMT                  0
Hasil - Lingkar Perut          0
Tajam Penglihatan            0
Tajam Pendengaran            0
Risiko Katarak               0
Skor PUMA                   0
dtype: int64

```

Gambar 4. 6 Hasil Cleaning data

4.3 Transformasi Data

Transformasi data adalah proses mengubah format atau struktur agar sesuai untuk analisis lebih lanjut. Ini termasuk mengubah tipe data kategori menjadi numerikal. Proses tersebut dapat dilihat pada gambar 4.7 sebagai berikut:

```

import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Load dataset dari path yang diberikan
file_path = '../content/dataset_cleaning.csv'
df_segmentation = pd.read_csv(file_path)

# Buat salinan dataset agar tidak memodifikasi data asli
df_encoded = df_segmentation.copy()

# Daftar kolom kategori yang ingin diubah
categorical_columns = ['JK', 'Tajam Penglihatan', 'Tajam Pendengaran', 'Risiko Katarak']

# Ubah kolom kategori menjadi numerik dan simpan mapping
label_encoders = {}
for column in categorical_columns:
    if column in df_encoded.columns:
        le = LabelEncoder()
        df_encoded[column] = le.fit_transform(df_encoded[column])
        label_encoders[column] = le # Simpan LabelEncoder untuk kolom ini

```

```

# Simpan DataFrame yang sudah dimodifikasi ke dalam file CSV
output_path = '/content/dataset_transformasi.csv'
df_encoded.to_csv(output_path, index=False)

# Tampilkan hasil transformasi untuk kolom yang di-encode
print("\nHasil transformasi untuk kolom 'JK', 'Tajam Penglihatan', 'Tajam Pendengaran', 'Risiko Katarak:')
for column in categorical_columns:
    if column in df_encoded.columns:
        print(f"\nDistribusi kategori untuk kolom '{column}':")
        print(df_encoded[column].value_counts())

# Tampilkan mapping dari kategori asli ke nilai numerik
mapping = {index: label for index, label in enumerate(label_encoders[column].classes_)}
print(f"\nMapping untuk kolom '{column}':")
for num, cat in mapping.items():
    print(f"{num}: {cat}")

```

Gambar 4. 7 Proses Transformasi data

4.3.1 Transformasi Jenis Kelamin

Pada atribut JK (Jenis Kelamin) dilakukan inisialisasi berdasarkan jenis kelamin *female* atau *male*.

Tabel 4. 2 Transformasi Jenis Kelamin

Jenis Kelamin	Nilai Transformasi	Frekuensi
female	0	7249
male	1	10067

4.3.2 Transformasi Tajam Penglihatan

Pada atribut Tajam Penglihatan dilakukan inisialisasi pada label gangguan berat, gangguan ringan, dan normal.

Tabel 4. 3 Transformasi Tajam Penglihatan

Tajam Penglihatan	Nilai Transformasi	Frekuensi
Gangguan Berat	0	1106
Gangguan Ringan	1	23
Normal	2	16187

4.3.3 Transformasi Tajam Pendengaran

Pada atribut Tajam Pendengaran dilakukan inisialisasi pada label gangguan pendengaran, normal.

Tabel 4. 4 Transformasi Tajam Pendengaran

Tajam Pendengaran	Nilai Transformasi	Frekuensi
Gangguan Pendengaran	0	19
Normal	1	17297

4.3.4 Transformasi Risiko Katarak

Pada atribut Risiko Katarak dilakukan inisialisasi pada label curiga katarak, gangguan penglihatan, dan normal.

Tabel 4. 5 Transformasi Risiko Katarak

Risiko Katarak	Nilai Transformasi	Frekuensi
Curiga Katarak	0	2
Gangguan Penglihatan	1	15
Normal	2	17299

4.4 K-Means Clustering

4.4.1 Menentukan titik pusat *cluster*

Pada tahap ini, penentuan titik cluster dilakukan untuk mengidentifikasi lokasi awal dari masing-masing cluster dalam analisis K-Means. Dalam penelitian ini, metode elbow digunakan untuk menentukan jumlah cluster yang optimal dengan tools Google Collab. Proses tersebut dapat dilihat pada gambar 4.8:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

# Load dataset dari path yang diberikan
file_path = '/content/dataset_transforsi1.csv'
df = pd.read_csv(file_path)

# Tentukan fitur yang akan digunakan untuk klustering
# Misalkan kita menggunakan semua kolom numerik
features = df.select_dtypes(include=['float64', 'int64'])

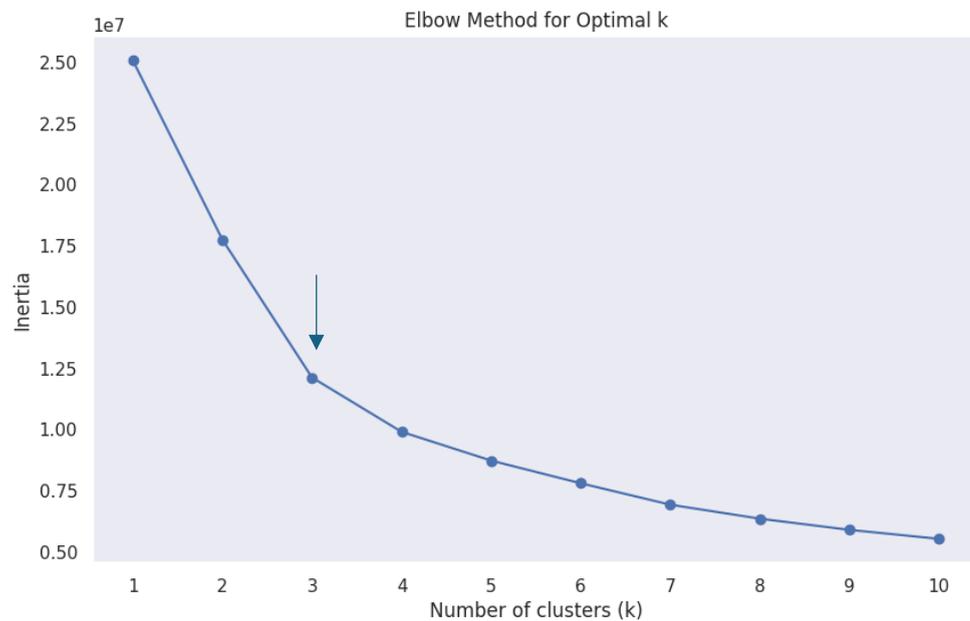
# Menentukan jumlah kluster yang akan diuji
inertia = []
k_values = range(1, 11) # Menguji dari 1 hingga 10 kluster

for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(features)
    inertia.append(kmeans.inertia_)

# Memplot hasilnya
plt.figure(figsize=(10, 6))
plt.plot(k_values, inertia, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of clusters (k)')
plt.ylabel('Inertia')
plt.xticks(k_values)
plt.grid()
```

Gambar 4. 8 Proses menentukan titik pusat cluster

Berikut hasil dari proses menentukan titik cluster dapat dilihat pada gambar 4.9:



Gambar 4. 9 Hasil titik pusat cluster

Berdasarkan visualisasi pada elbow method di atas menunjukkan bahwa kelengkungan tertajam sehingga membentuk sudut siku berada pada klaster 3. Sehingga ditentukan bahwa analisis klaster terbaik berada pada nilai $K=3$, jumlah dari setiap klaster dapat dilihat pada gambar 4.10:

```
# Jika ingin menghitung jumlah setiap kategori
kategori_counts = data_for_clustering['Cluster'].value_counts()
print("\nJumlah setiap kategori:")
kategori_counts.head()
```

Jumlah setiap kategori:

	count
Cluster	
2	7514
0	5561
1	4239

dtype: int64

Gambar 4. 10 Jumlah setiap klaster

4.4.2 Menentukan *Centroid* awal

Centroid awal adalah titik pusat dari masing-masing *cluster* yang ditentukan secara acak dari data training. Pemilihan titik *centroid* awal memiliki peran penting karena dapat memengaruhi hasil akhir dari proses pengelompokan *cluster*. Berikut adalah *centroid* awal yang digunakan dalam penelitian ini pada tabel 4.6:

Tabel 4. 6 Centroid awal

	Usia	JK	Hasil - Tekanan Darah (Sistol)	Hasil - Tekanan Darah (Diastol)	Hasil - Gula Darah Sewaktu	Hasil - IMT	Hasil - Lingkar Perut	Tajam Penglihatan	Tajam Pendengaran	Risiko Katarak	Skor PUMA
C0	35	0	110	70	100	19,1	73	2	1	2	2
C1	42	1	140	90	109	26,6	85	2	1	2	5
C2	42	0	150	90	200	24,4	80	2	1	2	4

4.4.3 Menghitung jarak data ketitik pusat cluster

Selanjutnya penulis melakukan perhitungan iterasi 1, dengan menghitung jarak seluruh data ke titik pusat *cluster* dengan rumus sebagai berikut :

$$De = \sqrt{(52 - 35)^2 + (1 - 0)^2 + \dots + (4 - 2)^2}$$

Berikut adalah contoh perhitungan data yang telah dilakukan iterasi pertama pada tabel 4.7:

Tabel 4. 7 Iterasi ke-1

Usia	JK	Hasil - Tekanan Darah (Sistol)	Hasil - Tekanan Darah (Diastol)	Hasil - Gula Darah Sewaktu	Hasil - IMT	Hasil - Lingkar Perut	Tajam Penglihatan	Tajam Pendengaran	Risiko Katarak	Skor PUMA	C0	C1	C2	Jarak Terdekat	Cluster
52	1	180	100	100	23,8	65	2	1	2	4	78,61355	47,85227	106,4254	47,8522727	1
44	1	110	90	325	23,4	60	2	1	2	4	226,4917	219,5364	132,7818	132,781776	2
52	1	180	100	100	23,8	65	2	1	2	3	78,59447	47,88361	106,4301	47,8836089	1
36	1	110	90	99	23,4	60	2	1	2	3	24,32057	40,9297	110,6345	24,3205674	0
36	1	110	90	99	26,5	60	0	1	2	7	25,60781	40,85352	110,7042	25,6078113	0
47	1	110	70	110	26	80	2	1	2	4	18,59059	36,77445	100,6407	18,590589	0
56	0	140	80	250	26,7	88	2	1	2	7	155,7201	142,0951	54,53705	54,5370516	2
44	1	110	90	325	23,4	60	2	1	2	4	226,4917	219,5364	132,7818	132,781776	2
50	1	140	90	230	22,2	70	1	1	2	4	135,8293	122,2758	34,21754	34,2175394	2
47	1	160	100	100	23,5	73	2	1	2	3	59,71064	27,63349	101,3746	27,6334942	1

4.4.4 Menghitung titik pusat *Centroid* baru

Pada iterasi pertama data telah menghasilkan keanggotaan *cluster*, selanjutnya penulis melakukan perhitungan untuk menentukan nilai titik pusat

cluster baru yang diperoleh dari menghitung rata-rata keanggotaan dari C0, C1 dan C2, berikut merupakan nilai centroid baru pada tabel 4.8:

Tabel 4. 8 Titik pusat cluster iterasi ke-2

	Usia	JK	Hasil - Tekanan Darah (Sistol)	Hasil - Tekanan Darah (Diastol)	Hasil - Gula Darah Sewaktu	Hasil - IMT	Hasil - Lingkar Perut	Tajam Penglihatan	Tajam Pendengaran	Risiko Katarak	Skor PUMA
C0	39,66667	1	110	83,33333	102,6667	25,3	66,66667	1,3333333	1	2	4,666667
C1	37,75	0,75	130	75	75	17,775	50,75	1,5	0,75	1,5	2,5
C2	48,5	0,75	125	87,5	282,5	23,925	69,5	1,75	1	2	4,75

Proses perhitungan nilai *centroid* yang baru telah dilakukan. Langkah selanjutnya yaitu menghitung kembali data dengan nilai *centroid* baru. Berikut merupakan tabel perhitungan iterasi ke-2 pada tabel 4.9:

Tabel 4. 9 Iterasi ke-2

	JK	Hasil - Tekanan Darah (Sistol)	Hasil - Tekanan Darah (Diastol)	Hasil - Gula Darah Sewaktu	Hasil - IMT	Hasil - Lingkar Perut	Tajam Penglihatan	Tajam Pendengaran	Risiko Katarak	Skor PUMA	C0	C1	C2	Jarak Terdekat	Cluster
52	1	180	100	100	23,8	65	2	1	2	4	73,09526	7,401877	191,1039	7,401877	1
44	1	110	90	325	23,4	60	2	1	2	4	222,5854	234,1699	46,35691	46,35691	2
52	1	180	100	100	23,8	65	2	1	2	3	73,11122	7,379325	191,1104	7,379325	1
36	1	110	90	99	23,4	60	2	1	2	3	11,07294	66,15538	184,8066	11,07294	0
36	1	110	90	99	26,5	60	0	1	2	7	11,15527	66,34469	184,8373	11,15527	0
47	1	110	70	110	26	80	2	1	2	4	21,55203	71,97269	174,3703	21,55203	0
56	0	140	80	250	26,7	88	2	1	2	7	152,8037	156,4612	41,82569	41,82569	2
44	1	110	90	325	23,4	60	2	1	2	4	222,5854	234,1699	46,35691	46,35691	2
50	1	140	90	230	22,2	70	1	1	2	4	131,4773	134,6119	54,71895	54,71895	2
47	1	160	100	100	23,5	73	2	1	2	3	53,71443	14,74735	186,2938	14,74735	1

Pada tahapan menghitung titik pusat centroid baru pada hasil dari iterasi kedua menunjukkan bahwa jumlah cluster yang terbentuk telah mencapai stabilitas. Setelah melakukan pengalokasian data ke dalam cluster berdasarkan jarak terdekat dan memperbarui posisi centroid, tidak ada perubahan signifikan pada lokasi centroid dibandingkan dengan iterasi sebelumnya. Hal ini menandakan bahwa pengelompokan data telah mencapai konvergensi, sehingga jumlah cluster yang dihasilkan dapat dianggap optimal untuk analisis lebih lanjut.

4.5 Interpretation / Evaluation

Data yang digunakan dalam penelitian ini merupakan data pasien UPT Puskesmas Sumur Batu. Adapun keseluruhan data berjumlah 17.316 pasien dengan pengujian menggunakan tools *Google Collab* dan *Tableu*.

4.5.1 Menampilkan Clustering hasil K-Means (PCA 2D)

Pada tahap ini berfungsi untuk memvisualisasikan hasil pengelompokan data kedalam tiga cluster menggunakan algoritma K-Means yang direduksi ke dua dimensi dengan Teknik Principal Component Analysis (PCA). Proses tersebut dapat dilihat pada gambar 4.11 sebagai berikut:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

# 1. Membaca dataset
df = pd.read_csv('/content/dataset_transforsi1.csv') # Pastikan path sesuai

# 2. Memilih kolom yang relevan
data_cleaned = df[['Hasil - Tekanan Darah (Sistol)', 'Hasil - Tekanan Darah (Diastol)']]

# 3. Menghapus nilai kosong atau string tidak valid
data_cleaned = data_cleaned.replace(' ', None) # Mengganti string kosong dengan None
data_cleaned = data_cleaned.dropna() # Menghapus baris dengan nilai NaN
data_cleaned = data_cleaned.astype(float) # Mengonversi data ke tipe float

# 4. Normalisasi data
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data_cleaned)

# 5. K-Means Clustering
kmeans = KMeans(n_clusters=3, random_state=42)
df_cleaned = data_cleaned.copy() # Salin data bersih untuk hasil clustering
df_cleaned['Cluster'] = kmeans.fit_predict(data_scaled)

# 6. Definisikan mapping cluster ke kategori hipertensi
cluster_to_category = {
    0: 'Normal',
    1: 'Pre-Hipertensi',
    2: 'Hipertensi'
}
```

```

# 7. Visualisasi hasil clustering (PCA untuk 2D)
pca = PCA(n_components=2)
data_pca = pca.fit_transform(data_scaled)

colors = ['yellow', 'blue', 'red']

plt.figure(figsize=(8, 6))

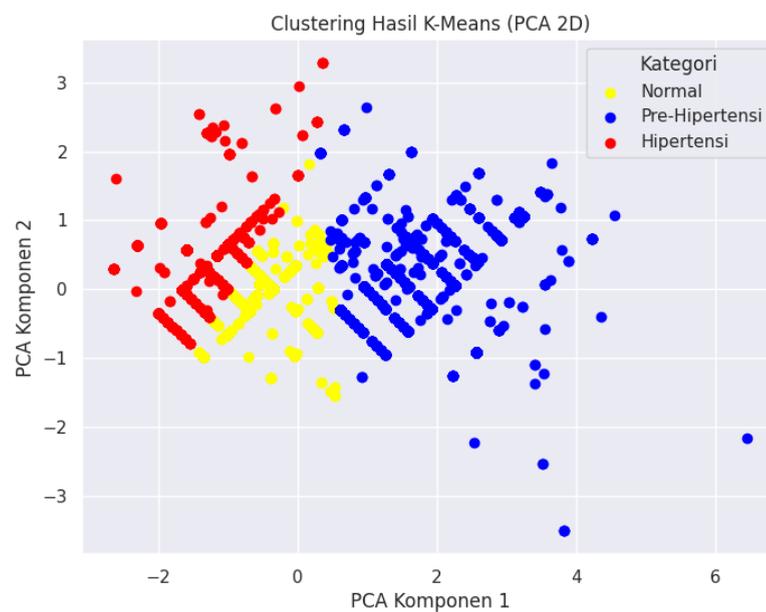
# Plot setiap kategori dengan warna dan label berbeda
for cluster_id, color in zip(cluster_to_category.keys(), colors):
    mask = df_cleaned['Cluster'] == cluster_id
    plt.scatter(data_pca[mask, 0], data_pca[mask, 1], label=cluster_to_category[cluster_id], color=color)

# Tambahkan keterangan dan judul
plt.title("Clustering Hasil K-Means (PCA 2D)")
plt.xlabel('PCA Komponen 1')
plt.ylabel('PCA Komponen 2')
plt.legend(title="Kategori")
plt.show()

```

Gambar 4. 11 Proses hasil K-Means (PCA 2D)

Berikut merupakan grafik clustering hasil K-means (PCA 2D), dapat dilihat pada gambar 4.12 sebagai berikut:



Gambar 4. 12 hasil K-Means (PCA 2D)

Sebagai hasil dari penelitian ini, proses hasil *K-Means Clustering* menggunakan Tableau dapat diakses melalui tautan berikut:

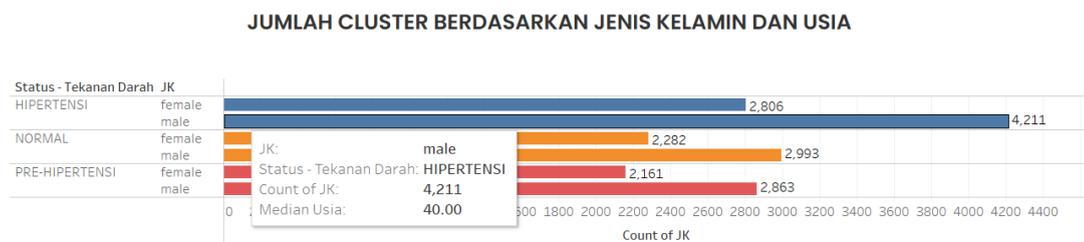
<https://colab.research.google.com/drive/1cG9IUB6yOZXXY-tF-eE3WK1MsG4TBRqX?usp=sharing>

Visualisasi ini mempermudah analisis pola dan distribusi cluster, sehingga karakteristik masing-masing cluster dapat dilihat dengan lebih jelas. Dalam penelitian ini, hasil menunjukkan bahwa data pasien berhasil dikelompokkan secara optimal ke dalam tiga cluster, yang dapat digunakan untuk memahami hubungan antara hipertensi dengan faktor risiko lainnya.

4.5.2 Pengujian *Tableau*

Tahap akhir yang dilakukan yaitu tahap pembuatan dashboard dan visualisasi data. Pada tahap ini, data yang sudah siap dapat langsung divisualisasikan menggunakan *Tableau Public* sehingga diperoleh sejumlah informasi yang diinginkan. Hal ini bertujuan untuk membantu dan mempermudah dalam proses dalam mengambil sebuah keputusan.

a. Visual Jumlah *Cluster* Berdasarkan Jenis Kelamin dan Usia

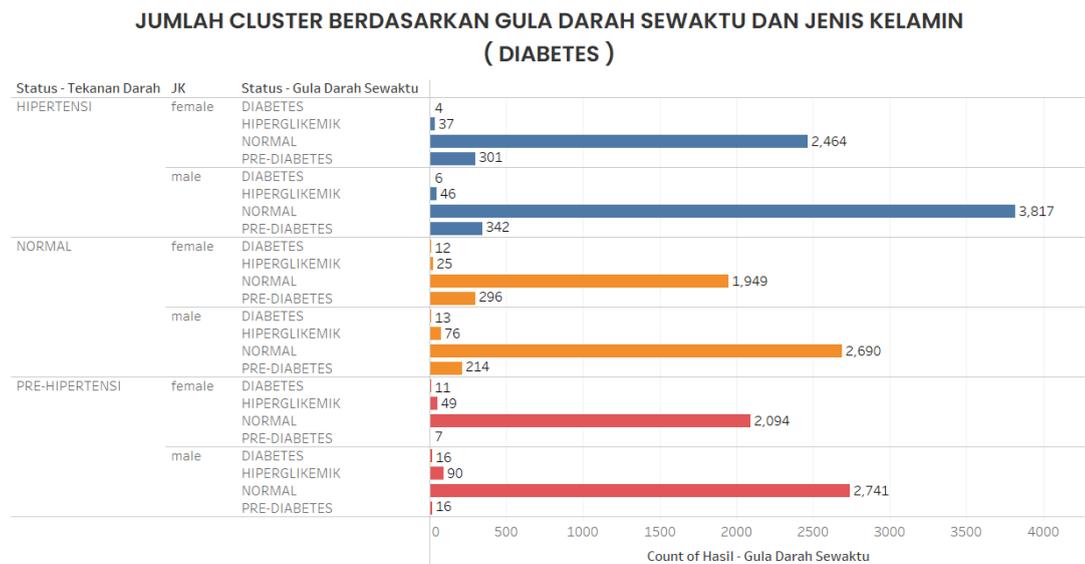


Gambar 4. 13 Visualisasi Jumlah *Cluster* Berdasarkan jenis kelamin dan usia

Dapat dilihat pada gambar 4.13 Jumlah *Cluster* berdasarkan jenis kelamin dan usia terbanyak ada pada Status tekanan darah hipertensi yang berjenis kelamin pria dengan rata-rata usia 40 tahun sebanyak 4.211 pasien, wanita dengan rata-rata usia 42 tahun sebanyak 2.806 pasien, pada Status tekanan darah pre-hipertensi yang berjenis kelamin pria dengan rata-rata usia 31 tahun sebanyak 2.863 pasien, wanita dengan rata-rata usia 32 tahun sebanyak

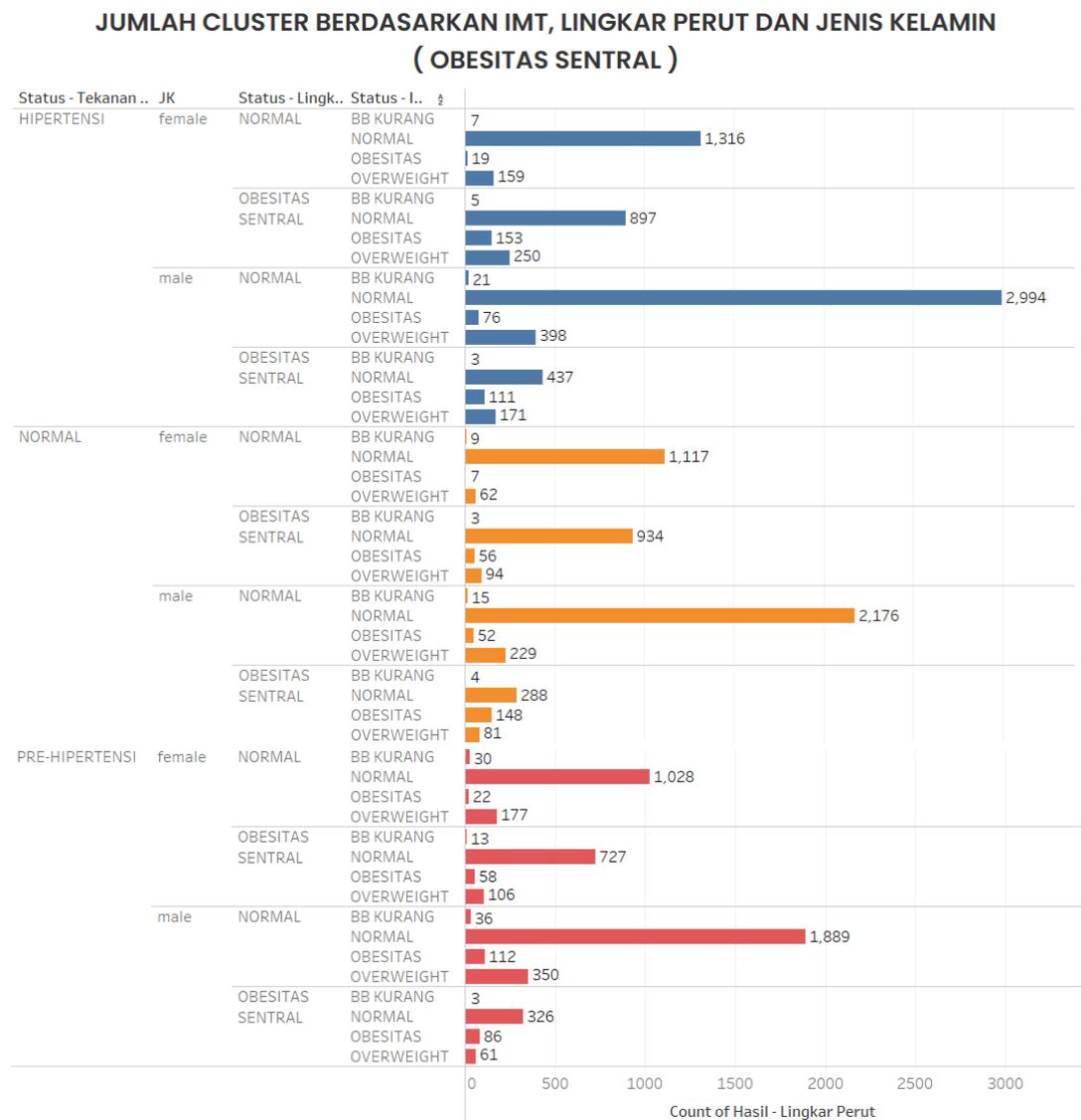
2.161 pasien, pada Status tekanan darah normal yang berjenis kelamin pria dengan rata-rata usia 32 tahun sebanyak 2.806 pasien, wanita dengan rata-rata usia 37 tahun sebanyak 2.806 pasien.

b. Visual Jumlah *Cluster* Berdasarkan Risiko Penyakit Diabetes



Gambar 4. 14 Visualisasi Jumlah Cluster Berdasarkan Risiko Penyakit Diabetes

Dapat dilihat pada gambar 4.14 Jumlah *Cluster* Berdasarkan Risiko Penyakit Diabetes, berdasarkan status gula darah sewaktu, pada status tekanan darah hipertensi jenis kelamin wanita yang mengalami diabetes sebanyak 4 pasien dengan rata-rata usia 42 tahun dan pria 6 pasien dengan rata-rata usia 40 tahun, berdasarkan status tekanan darah pre-hipertensi jenis kelamin wanita 11 pasien dengan rata-rata usia 40 tahun dan pria 16 pasien dengan rata-rata usia 40 tahun, berdasarkan status tekanan darah normal jenis kelamin wanita 12 pasien dengan rata-rata usia 40 tahun dan pria 13 pasien dengan rata-rata usia 41 tahun.

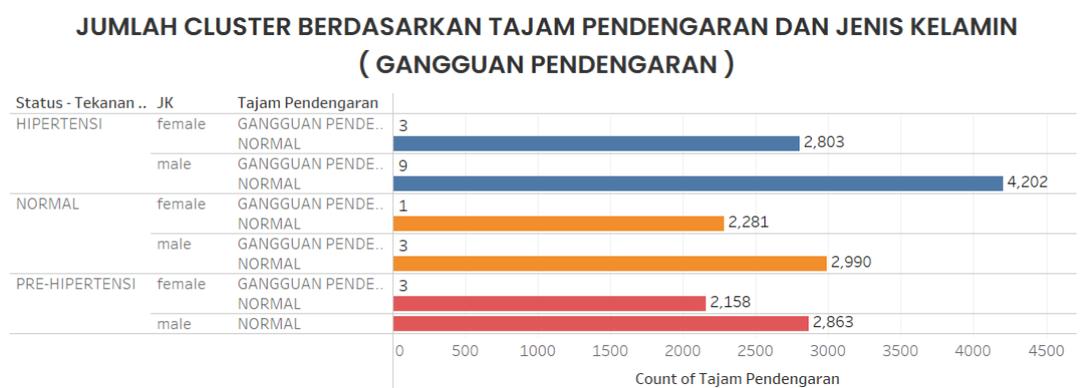
c. Visual Jumlah *Cluster* Berdasarkan Risiko Penyakit Obesitas Sentral

Gambar 4. 16 Visualisasi Jumlah Cluster Berdasarkan Risiko Penyakit Obesitas

Dapat dilihat pada gambar 4.16 Jumlah *Cluster* Berdasarkan Risiko Penyakit Obesitas Sentral, berdasarkan status lingkar perut dan indeks massa tubuh (IMT), pada status tekanan darah hipertensi wanita yang mengalami obesitas sentral sebanyak 153 pasien dengan rata-rata usia 43 tahun dan pria 111 pasien dengan rata-rata usia 49 tahun, pada status tekanan darah pre-hipertensi wanita sebanyak 58 pasien dengan rata-rata

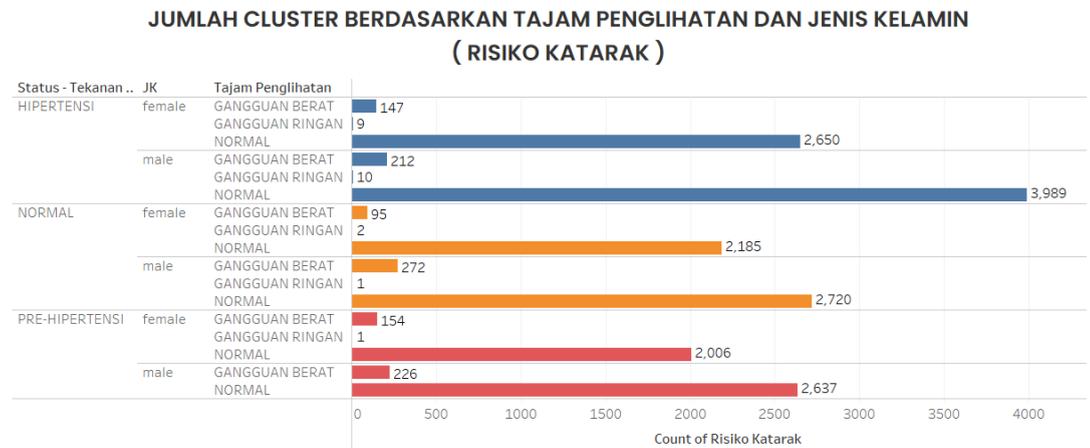
usia 24 tahun dan pria sebanyak 86 pasien dengan rata-rata usia 24 tahun, pada status tekanan darah normal wanita sebanyak 52 pasien dengan rata-rata usia 25 tahun dan jenis kelamin pria sebanyak 148 pasien dengan rata-rata usia 25 tahun.

d. Visual Jumlah *Cluster* Berdasarkan Risiko Penyakit Gangguan Pendengaran



Gambar 4. 17 Visualisasi Jumlah Cluster Berdasarkan Risiko Penyakit Gangguan Pendengaran

Dapat dilihat pada gambar 4.17 Jumlah *Cluster* Berdasarkan Risiko Penyakit Gangguan Pendengaran, berdasarkan tajam pendengaran yang mengalami gangguan pendengaran pada status tekanan darah hipertensi wanita 3 pasien dengan rata-rata usia 51 tahun, dan pria 9 pasien dengan rata-rata usia 28 tahun, pada status tekanan darah pre-hipertensi wanita 3 pasien dengan rata-rata usia 40 tahun, pada status tekanan darah normal wanita 1 pasien dengan rata-rata usia 17 tahun dan pria 3 pasien dengan rata-rata usia 34 tahun.

e. Visual Jumlah *Cluster* Berdasarkan Risiko Penyakit Risiko Katarak

Gambar 4. 18 Visualisasi Jumlah Cluster Berdasarkan Risiko Penyakit Risiko Katarak

Dapat dilihat pada gambar 4.18 Jumlah *Cluster* Berdasarkan Risiko Penyakit Risiko Katarak, berdasarkan risiko katarak dengan gejala gangguan berat pada status tekanan darah hipertensi wanita sebanyak 147 pasien dengan rata-rata usia 42 tahun dan pria 212 pasien dengan rata-rata usia 42 tahun, pada status tekanan darah pre-hipertensi wanita 154 pasien dengan rata-rata usia 25 tahun dan pria 226 pasien dengan rata-rata usia 22 tahun, pada status tekanan darah normal wanita 95 pasien dengan rata-rata usia 29 tahun dan pria 272 pasien dengan rata-rata usia 29 tahun.

- f. Visual Jumlah *Cluster* berdasarkan Risiko Penyakit Paru Obstruktif Kronis (PPOK)



Gambar 4. 19 Visualisasi Jumlah Cluster Berdasarkan Risiko Penyakit Paru Obstruktif Kronis (PPOK)

Dapat dilihat pada gambar 4.19 Visualisasi *Cluster* Berdasarkan Risiko Penyakit Paru Obstruktif Kronis (PPOK), berdasarkan status tekanan darah hipertensi dengan status puma risiko tinggi PPOK pada wanita sebanyak 104 pasien dengan rata-rata usia 47 tahun dan pria 187 pasien dengan rata-rata usia 49 tahun, pada status tekanan darah pre-hipertensi wanita 57 pasien dengan rata-rata usia 31 tahun dan pria 90 pasien dengan rata-rata usia 29 tahun, pada status tekanan darah normal wanita 99 pasien dengan rata-rata usia 43 tahun dan pria 98 pasien dengan rata-rata usia 38 tahun.

- g. Visual Jumlah *cluster* berdasarkan keseluruhan fitur.



Gambar 4. 20 Visualisasi Jumlah *Cluster* Berdasarkan Keseluruhan Fitur

Pada gambar 4.20 Visualisasi Jumlah *Cluster* Berdasarkan Keseluruhan Fitur menunjukkan distribusi jumlah cluster berdasarkan keseluruhan fitur yang digunakan dalam analisis. Sumbu horizontal merepresentasikan jumlah individu (Count of JK), sedangkan sumbu vertikal menunjukkan kategori Status Tekanan Darah, yang terbagi menjadi Hipertensi, Normal, dan Pre-Hipertensi. Pada visualisasi tersebut, warna biru dan oranye menggambarkan segmentasi data berdasarkan kategori jenis kelamin. Berdasarkan visualisasi yang ditampilkan, salah satu *cluster* menunjukkan individu dengan karakteristik berikut, dapat dilihat pada gambar 4.21:

JK:	male
Status - Tekanan Darah:	HIPERTENSI
Count of JK:	4,211
Distinct count of Skor PUMA:	8
Distinct count of Tajam Pendengaran:	2
Distinct count of Tajam Penglihatan:	3
Distinct count of Usia:	69
Median Hasil - Gula Darah Sewaktu:	100.00
Median Hasil - IMT:	23.100
Median Hasil - Lingkar Perut:	80.00

Gambar 4. 21 Hasil Visualisasi Jumlah *Cluster* Berdasarkan Keseluruhan Fitur

Dari visualisasi ini, dapat dianalisis bahwa jumlah individu dengan kategori Hipertensi cukup tinggi, dengan distribusi yang lebih besar dibandingkan kategori lainnya. Sebagai hasil dari penelitian ini, visualisasi data yang dihasilkan menggunakan *Tableau* dapat diakses melalui tautan berikut:

https://public.tableau.com/views/30_17382477128780/Dashboard1?:language=en-US&publish=yes&:sid=&:redirect=auth&:display_count=n&:origin=viz_share_link

Visualisasi ini dapat membantu pemahaman lebih lanjut mengenai pola hipertensi serta menjadi referensi bagi peneliti selanjutnya dalam mengembangkan model yang lebih akurat dan aplikatif.