**Mata Kuliah Analisis Multivariat**

**Tugas Clustering KMeans Pertemuan 11**

**Nama Kelompok:**

* Fadilah Ratu Azzahra (5231811015)
* Lathif Ramadhan (5231811022)

**Deskripsi Dataset**

**Sumber Data** Dataset ini berasal dari *National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK)*, berisi rekam medis pasien perempuan keturunan Indian Pima berusia ≥21 tahun. Tujuan utama pengumpulan data ini adalah untuk memprediksi diabetes berdasarkan parameter klinis tertentu.

Link sumber dataset: <https://www.kaggle.com/datasets/whenamancodes/predict-diabities>

**Variabel yang Digunakan** Dalam analisis klastering ini, dua variabel digunakan sebagai atribut pembentuk klaster:

1. **BloodPressure (Tekanan Darah)** Mengukur tekanan darah diastolik pasien (dalam satuan mmHg). Nilai ini merefleksikan kondisi kardiovaskular pasien, di mana pola abnormal dapat terkait dengan risiko gangguan metabolik seperti diabetes. Rentang nilai dalam dataset: 0–122 mmHg.
2. **Age (Usia)** Menunjukkan usia pasien (dalam tahun). Faktor usia merupakan prediktor signifikan dalam perkembangan diabetes, mengingat prevalensi kondisi ini meningkat seiring penuaan. Rentang usia dalam dataset: 21–81 tahun.

**Relevansi dalam Analisis Klastering** Pemilihan kedua variabel didasarkan pada:

* **Dampak fisiologis**: Tekanan darah dan usia memiliki korelasi klinis dengan resistensi insulin dan fungsi metabolik.
* **Pembedaan profil risiko**: Kombinasi variabel ini memungkinkan identifikasi kelompok pasien dengan karakteristik kesehatan serupa, seperti kelompok usia muda dengan hipertensi vs. kelompok lansia dengan tekanan darah stabil.
* **Efektivitas visualisasi**: Dimensi dua variabel memfasilitasi interpretasi pola klaster secara grafis.

**Signifikansi Analitis**

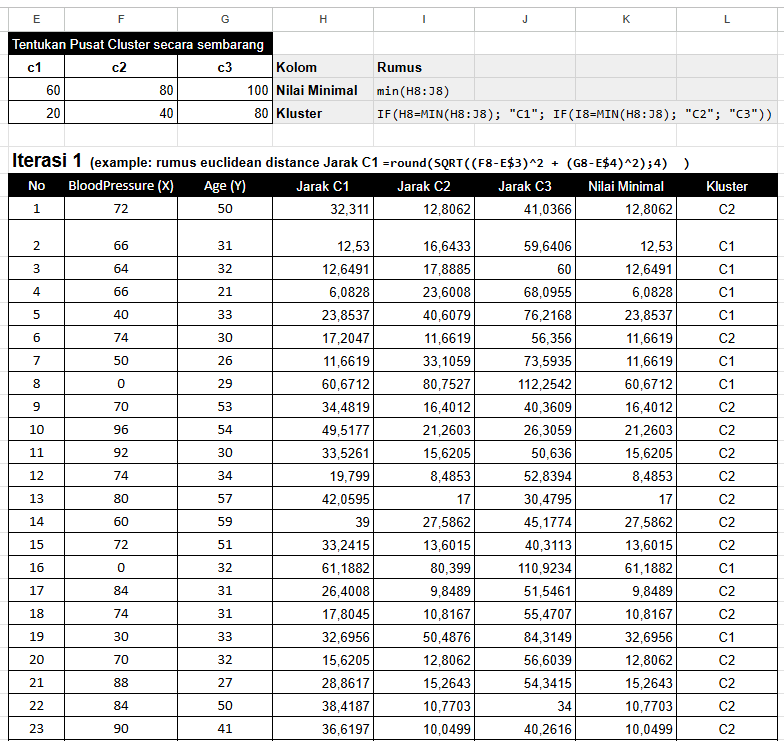
Pengelompokan berbasis tekanan darah dan usia membantu mengidentifikasi segmen pasien yang memerlukan intervensi spesifik. Contoh potensial temuan:

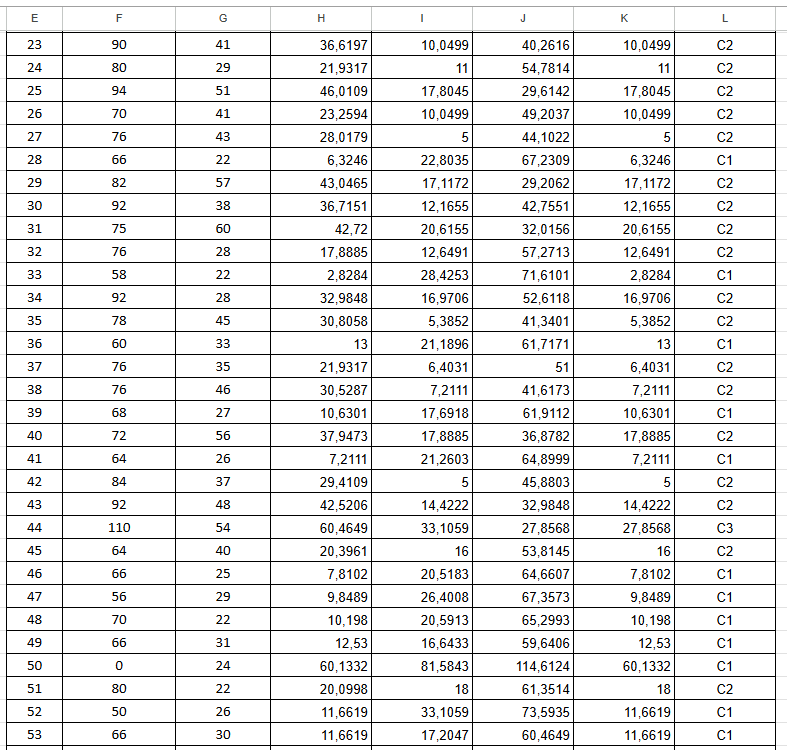
* Klaster usia produktif (<40 tahun) dengan tekanan darah tinggi, mengindikasikan risiko diabetes dini.
* Klaster lansia (>60 tahun) dengan tekanan darah normal, merepresentasikan kelompok dengan manajemen kesehatan lebih baik.

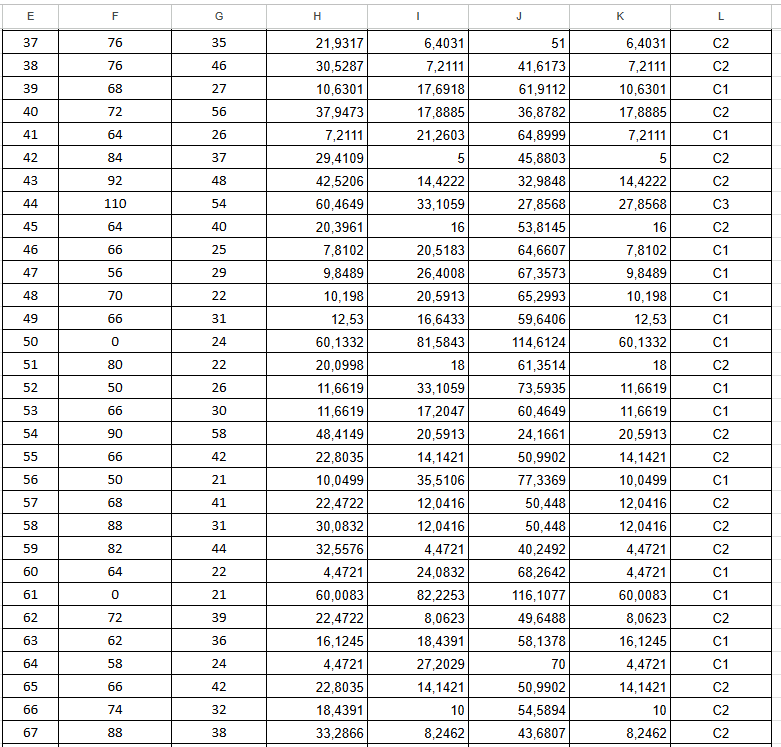
Analisis ini memberikan dasar stratifikasi risiko untuk rekomendasi kebijakan kesehatan yang lebih terarah.

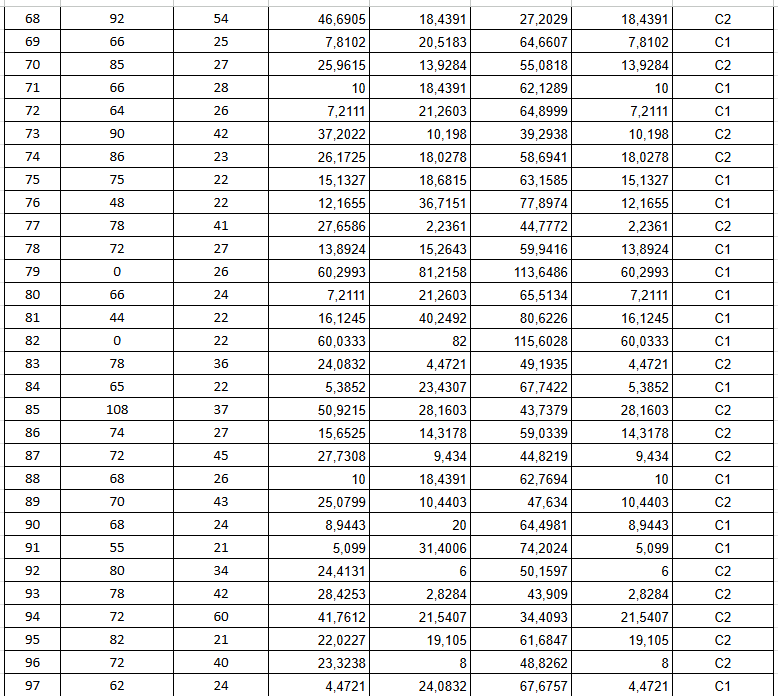
1. **Pengerjaan Manual Menggunakan Excell**

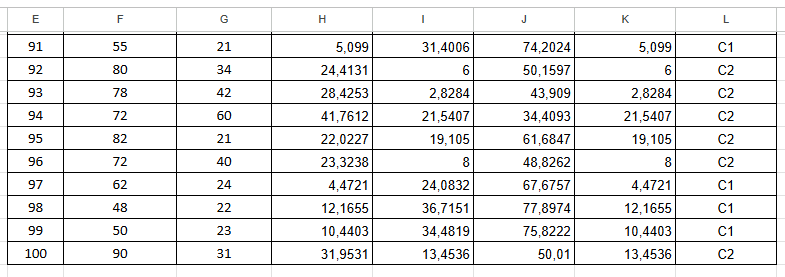
**Iterasi 1**

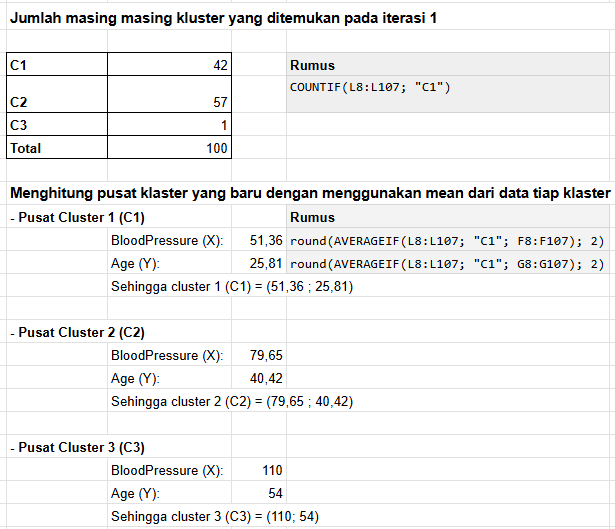




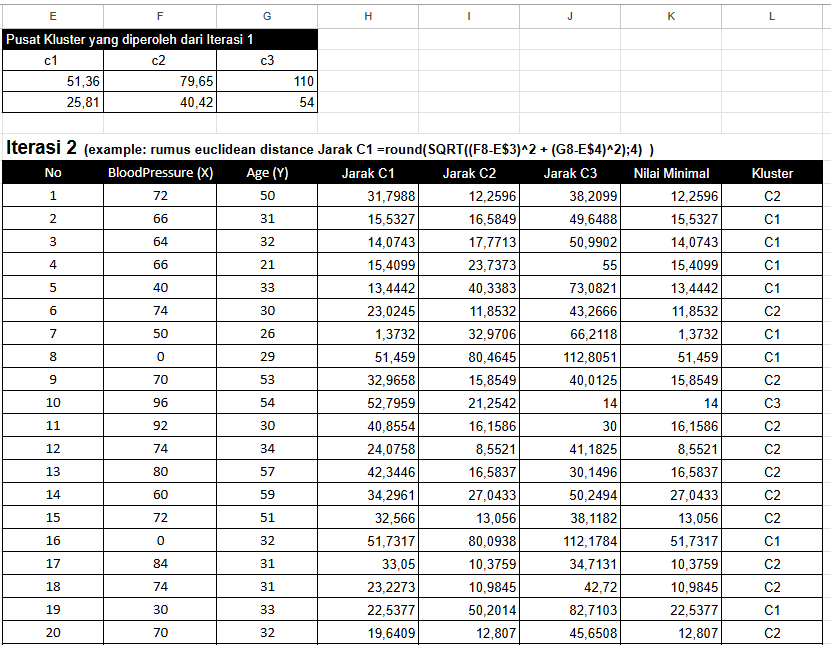








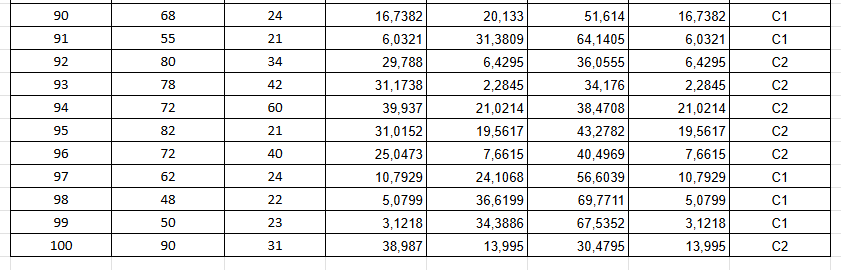
**Iterasi 2:**

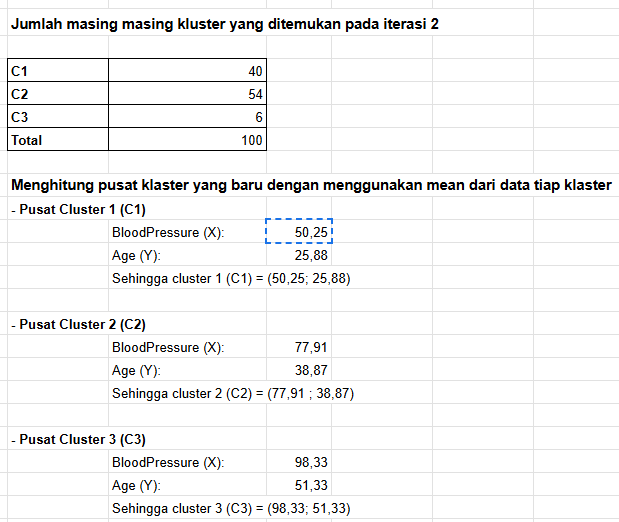


.

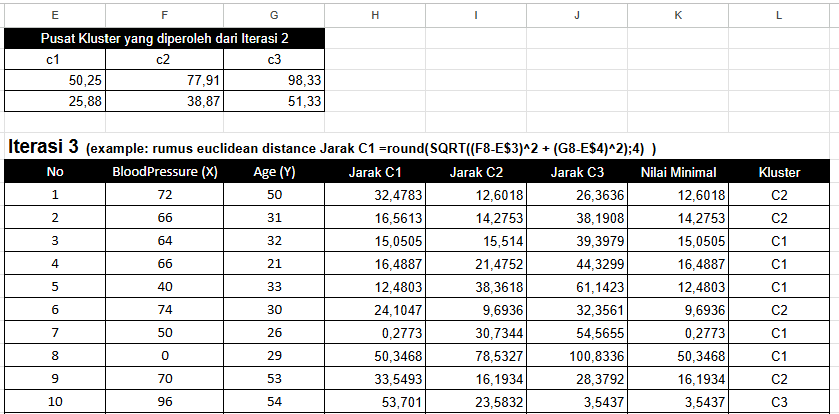
.

.



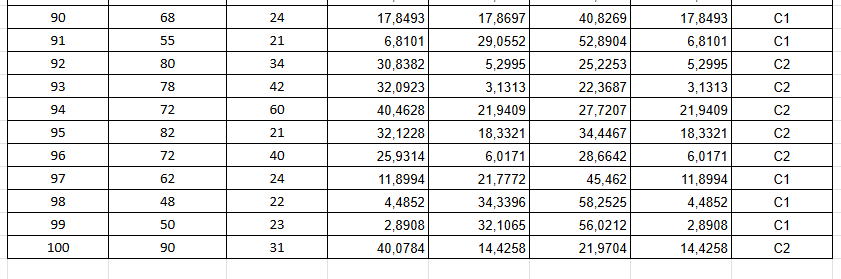


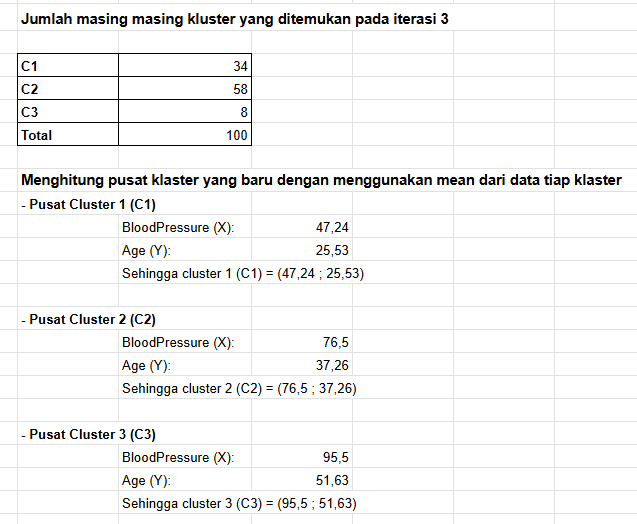
**Iterasi 3:**



.

.





**Kesimpulan Excel:**

Berdasarkan perhitungan iterasi ketiga, diperoleh hasil klastering sebagai berikut:

1. **Distribusi Pasien per Klaster**:
   * Klaster 1 (C1): **34 pasien**
   * Klaster 2 (C2): **58 pasien** (kelompok terbesar)
   * Klaster 3 (C3): **8 pasien**
2. **Profil Karakteristik Klaster**:
   * **C1**: Rata-rata tekanan darah **47,24 mmHg** dan usia **25,53 tahun**.  
      *(Profil: Pasien muda dengan tekanan darah rendah)*
   * **C2**: Rata-rata tekanan darah **76,50 mmHg** dan usia **37,26 tahun**.  
      *(Profil: Dewasa dengan tekanan darah normal)*
   * **C3**: Rata-rata tekanan darah **95,50 mmHg** dan usia **51,63 tahun**.  
      *(Profil: Lansia dengan tekanan darah tinggi)*
3. **Stabilitas Algoritma**:
   * Centroid tidak mengalami perubahan signifikan pada iterasi ketiga, menunjukkan **konvergensi telah tercapai**.

**Implikasi Analitis**:  
 Hasil ini mengidentifikasi tiga kelompok pasien dengan risiko klinis berbeda:

* **C1**: Potensi risiko gangguan metabolik pada usia muda.
* **C2**: Kelompok stabil dengan manajemen kesehatan memadai.
* **C3**: Kelompok prioritas intervensi akibat tekanan darah tinggi dan usia lanjut.

1. **Pengerjaan Menggunakan Python**

**Import Library**

import pandas as pd

import numpy as np

import kagglehub

import os

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

from sklearn.impute import SimpleImputer

from sklearn.metrics import silhouette\_score

**1. Load Dataset (kolom BloodPressure dan Age serta 100 baris pertama)**

path = kagglehub.dataset\_download('whenamancodes/predict-diabities')

print(f"Konten/dataset dari downloaded directory '{path}':")

for item in os.listdir(path):

print(item)

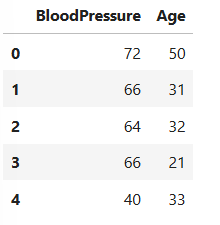
csv\_file\_path = os.path.join(path, 'diabetes.csv')

Konten/dataset dari downloaded directory '/kaggle/input/predict-diabities':

diabetes.csv

df = pd.read\_csv(csv\_file\_path, usecols=['BloodPressure', 'Age'], nrows=100)

df.head()



df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 100 entries, 0 to 99

Data columns (total 2 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

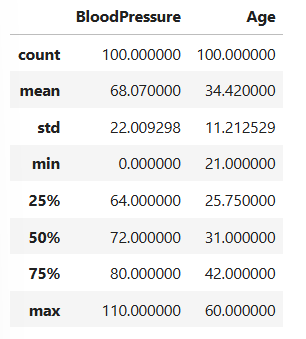
0 BloodPressure 100 non-null int64

1 Age 100 non-null int64

dtypes: int64(2)

memory usage: 1.7 KB

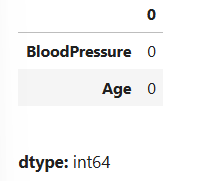
df.describe()



**2. Data Preprocessing**

Cek missing Values

df.isnull().sum()



*# Mengganti nilai 0 di BloodPressure dengan NaN*

df['BloodPressure'] = df['BloodPressure'].replace(0, np.nan)

In [8]:

*# Cek missing values setelah penggantian*

print("Missing values setelah mengganti 0 dengan NaN:")

print(df.isnull().sum())

Missing values setelah mengganti 0 dengan NaN:

BloodPressure 6

Age 0

dtype: int64

*# Imputasi missing values (menggunakan median agar tidak terlalu terpengaruh outlier jika ada)*

imputer = SimpleImputer(strategy='median')

df[['BloodPressure']] = imputer.fit\_transform(df[['BloodPressure']])

*# Cek kembali info dan describe setelah imputasi*

print("\nInfo setelah imputasi:")

df.info()

print("\nDescribe setelah imputasi:")

print(df.describe())

Info setelah imputasi:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 100 entries, 0 to 99

Data columns (total 2 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 BloodPressure 100 non-null float64

1 Age 100 non-null int64

dtypes: float64(1), int64(1)

memory usage: 1.7 KB

Describe setelah imputasi:

BloodPressure Age

count 100.000000 100.000000

mean 72.390000 34.420000

std 13.626248 11.212529

min 30.000000 21.000000

25% 66.000000 25.750000

50% 72.000000 31.000000

75% 80.000000 42.000000

max 110.000000 60.000000

Standarisasi data

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(df[['BloodPressure', 'Age']])

**3. Menentukan dan Melihat Jumlah Klaster Optimal (Elbow Metthod) secara otomatis**

* Menghitung SSE (Sum of Squared Errors) untuk k=1 hingga 10
* Mendeteksi titik siku secara otomatis

for k in range(1, 11):

kmeans = KMeans(n\_clusters=k, random\_state=42, n\_init=10)

kmeans.fit(scaled\_data)

sse.append(kmeans.inertia\_)

*# Visualisasi Elbow Method yang lebih informatif*

plt.figure(figsize=(12, 8))

k\_range = range(1, 11) *# Define k\_range*

*# Plot SSE vs k*

plt.plot(k\_range, sse, 'bo-', linewidth=2, markersize=8)

plt.title('Metode Elbow untuk Menentukan Jumlah Klaster Optimal', fontsize=16)

plt.xlabel('Jumlah Klaster (k)', fontsize=14)

plt.ylabel('Sum of Squared Errors (SSE)', fontsize=14)

plt.xticks(k\_range)

plt.grid(True, linestyle='--', alpha=0.7)

*# Tambahkan garis penanda dan anotasi*

for k in k\_range:

plt.annotate(f"k={k}\nSSE={sse[k-1]:.1f}",

(k, sse[k-1]),

textcoords="offset points",

xytext=(0,10),

ha='center',

fontsize=10)

*# Garis penanda khusus untuk titik-titik kunci*

plt.axvline(x=2, color='gray', linestyle=':', alpha=0.5)

plt.axvline(x=3, color='gray', linestyle=':', alpha=0.5)

plt.axvline(x=4, color='gray', linestyle=':', alpha=0.5)

plt.axvline(x=5, color='gray', linestyle=':', alpha=0.5)

*# Highlight area elbow point*

plt.axvspan(2, 4, color='yellow', alpha=0.1, label='Area Potensial Elbow Point')

*# Tambahkan panah untuk menunjukkan lokasi elbow*

plt.annotate('Titik Elbow Potensial',

xy=(3, sse[2]),

xytext=(4, sse[2] + 100),

arrowprops=dict(arrowstyle='->', color='red'),

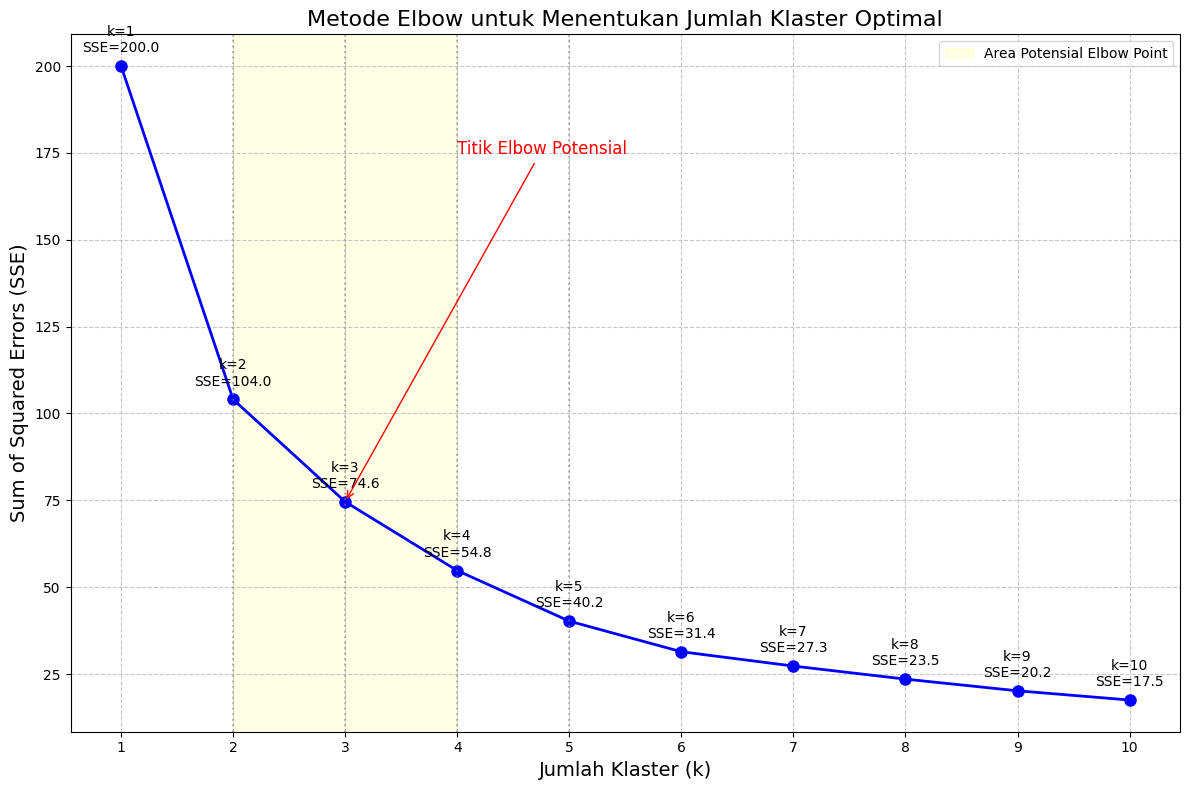
fontsize=12, color='red')

plt.legend()

plt.tight\_layout()

plt.savefig('elbow\_method\_detailed.png', dpi=300)

plt.show()



k\_optimal = int(input("\nMasukkan jumlah klaster (k) yang Anda pilih: "))

Masukkan jumlah klaster (k) yang Anda pilih: 3

**4. K-Means clustering**

* Membuat model dengan k optimal
* Menambahkan label klaster ke dataset

kmeans = KMeans(n\_clusters=k\_optimal, random\_state=42, n\_init=10)

df['Cluster'] = kmeans.fit\_predict(scaled\_data)

centroids = scaler.inverse\_transform(kmeans.cluster\_centers\_)

**5. Menghitung statistik per klaster**

cluster\_stats = df.groupby('Cluster').agg(

Patients=('Age', 'count'),

Avg\_Age=('Age', 'mean'),

Avg\_BP=('BloodPressure', 'mean'),

BP\_Std=('BloodPressure', 'std')

).reset\_index()

**6. Visualize results**

* Scatter plot dengan warna berbeda per klaster
* Menampilkan posisi centroid
* Garis referensi tekanan darah

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

colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728']

for i in range(k\_optimal):

cluster\_data = df[df['Cluster'] == i]

plt.scatter(cluster\_data['Age'], cluster\_data['BloodPressure'],

color=colors[i], alpha=0.7, s=50, label=f'Klaster {i}')

*# Plot centroid*

plt.scatter(centroids[:, 1], centroids[:, 0], marker='X', s=200,

color='black', label='Centroid')

*# Garis referensi medis*

plt.axhline(90, color='orange', linestyle='--', alpha=0.7, label='Batas Hipotensi')

plt.axhline(120, color='green', linestyle='--', alpha=0.7, label='Batas Normal')

*# Format visualisasi*

plt.title(f'Klastering K-Means (k={k\_optimal})\nTekanan Darah vs Usia', fontsize=14)

plt.xlabel('Usia (Tahun)', fontsize=12)

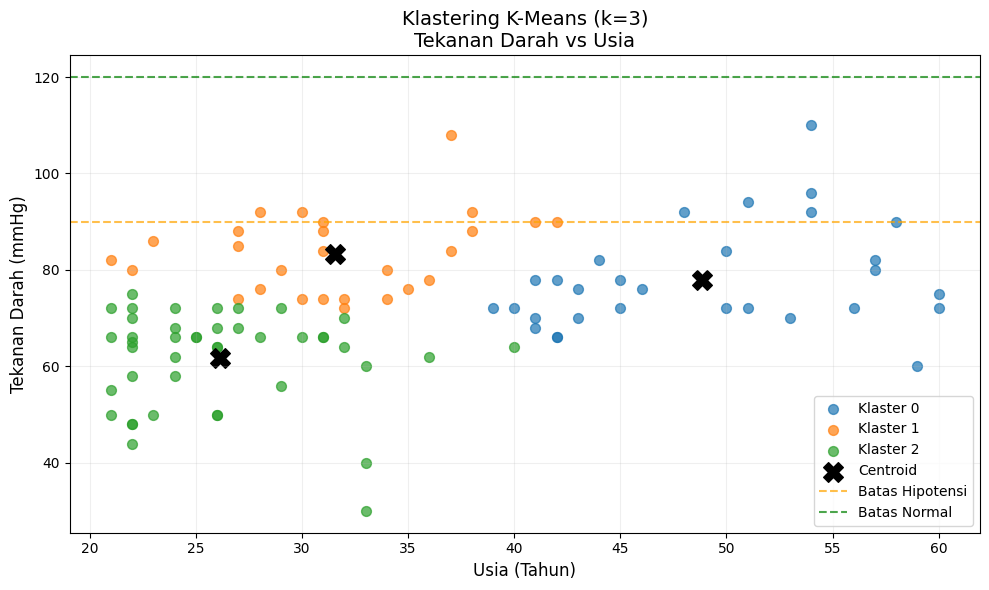
plt.ylabel('Tekanan Darah (mmHg)', fontsize=12)

plt.legend()

plt.grid(alpha=0.2)

plt.tight\_layout()

plt.savefig('hasil\_klastering.png', dpi=300)



**7. Laporan Hasil**

print("="\*50)

print(f"{'ANALISIS KLASTERING K-MEANS (200 baris pertama)':^50}")

print("="\*50)

print(f"{'Jumlah klaster optimal:':<25} {k\_optimal}")

print(f"{'Total pasien:':<25} {len(df)}")

print(f"{'Skor Silhouette:':<25} {silhouette\_score(scaled\_data, df['Cluster']):.3f}")

print("\n" + "-"\*50)

print(f"{'STATISTIK PER KLASTER':^50}")

print("-"\*50)

print(cluster\_stats.round(1).to\_string(index=False))

print("\n" + "-"\*50)

print(f"{'INTERPRETASI MEDIS':^50}")

print("-"\*50)

*# Interpretasi hasil berdasarkan standar medis*

for i in range(k\_optimal):

stats = cluster\_stats[cluster\_stats['Cluster'] == i].iloc[0]

*# Kategorisasi tekanan darah*

if stats['Avg\_BP'] < 90:

status\_td = "Hipotensi"

elif stats['Avg\_BP'] < 120:

status\_td = "Normal"

elif stats['Avg\_BP'] < 130:

status\_td = "Pra-hipertensi"

else:

status\_td = "Hipertensi"

*# Kategorisasi kelompok usia*

if stats['Avg\_Age'] < 30:

kelompok\_usia = "Muda"

elif stats['Avg\_Age'] < 50:

kelompok\_usia = "Dewasa"

else:

kelompok\_usia = "Lanjut Usia"

print(f"Klaster {i} ({int(stats['Patients'])} pasien):")

print(f" - Profil khas: Pasien {kelompok\_usia} dengan {status\_td}")

print(f" - Rerata Usia: {stats['Avg\_Age']:.1f} tahun | Rerata Tekanan darah: {stats['Avg\_BP']:.1f} mmHg")

==================================================

ANALISIS KLASTERING K-MEANS (200 baris pertama)

==================================================

Jumlah klaster optimal: 3

Total pasien: 100

Skor Silhouette: 0.368

--------------------------------------------------

STATISTIK PER KLASTER

--------------------------------------------------

Cluster Patients Avg\_Age Avg\_BP BP\_Std

0 30 48.9 77.9 10.8

1 27 31.6 83.4 8.3

2 43 26.1 61.7 9.9

--------------------------------------------------

INTERPRETASI MEDIS

--------------------------------------------------

Klaster 0 (30 pasien):

- Profil khas: Pasien Dewasa dengan Hipotensi

- Rerata Usia: 48.9 tahun | Rerata Tekanan darah: 77.9 mmHg

Klaster 1 (27 pasien):

- Profil khas: Pasien Dewasa dengan Hipotensi

- Rerata Usia: 31.6 tahun | Rerata Tekanan darah: 83.4 mmHg

Klaster 2 (43 pasien):

- Profil khas: Pasien Muda dengan Hipotensi

- Rerata Usia: 26.1 tahun | Rerata Tekanan darah: 61.7 mmHg

**Kesimpulan Hasil Analisis dengan Python** Proses klastering menggunakan Python menghasilkan tiga kelompok pasien sebagai berikut:

1. **Klaster 0** (30 pasien):
   * Rata-rata usia: **48,9 tahun** (kategori *Dewasa*)
   * Rata-rata tekanan darah: **77,9 mmHg** (kategori *Hipotensi*)
   * Standar deviasi tekanan darah: 10,8 mmHg
2. **Klaster 1** (27 pasien):
   * Rata-rata usia: **31,6 tahun** (kategori *Dewasa*)
   * Rata-rata tekanan darah: **83,4 mmHg** (kategori *Hipotensi*)
   * Standar deviasi tekanan darah: 8,3 mmHg
3. **Klaster 2** (43 pasien):
   * Rata-rata usia: **26,1 tahun** (kategori *Muda*)
   * Rata-rata tekanan darah: **61,7 mmHg** (kategori *Hipotensi*)
   * Standar deviasi tekanan darah: 9,9 mmHg

**Evaluasi Model**:

* Skor Silhouette: **0,368** (menunjukkan pemisahan klaster *cukup baik*).
* Semua klaster tergolong **hipotensi** berdasarkan klasifikasi medis (tekanan darah <90 mmHg).

**Temuan Kritis**:

* Terdapat perbedaan usia signifikan antar klaster, namun tekanan darah seluruh kelompok secara konsisten rendah.
* Klaster 0 (dewasa) dan Klaster 1 (dewasa muda) memiliki tekanan darah lebih tinggi dibanding Klaster 2 (muda), meski tetap dalam kategori hipotensi.

**Catatan Analitis**:  
 Hasil ini mengindikasikan bahwa faktor usia tidak secara langsung berkorelasi dengan peningkatan tekanan darah pada sampel data ini. Seluruh kelompok memerlukan pemantauan terkait risiko komplikasi akibat hipotensi.

**Penjelasan Tambahan**:

* **Kategori Usia**: Muda (<30 tahun), Dewasa (30-49 tahun), Lanjut Usia (≥50 tahun).
* **Kategori Tekanan Darah**: Hipotensi (<90 mmHg), Normal (90-119 mmHg), Pra-hipertensi (120-129 mmHg), Hipertensi (≥130 mmHg).
* **Skor Silhouette**: Rentang -1 hingga 1; semakin mendekati 1, semakin baik pemisahan klaster.

**Kesimpulan Akhir Laporan**

Berdasarkan analisis klastering K-Means terhadap 100 pasien perempuan keturunan Pima Indian, diperoleh temuan kunci sebagai berikut:

1. **Konsistensi Temuan Dua Metode**
   * Baik perhitungan manual di Excel maupun otomatisasi dengan Python berhasil mengidentifikasi **tiga kelompok pasien** dengan profil tekanan darah dan usia berbeda.
   * Excel menampilkan konvergensi algoritma dalam 3 iterasi, sementara Python mengonfirmasi k=3 sebagai jumlah klaster optimal melalui metode Elbow (Skor Silhouette: 0.368).
2. **Profil Risiko Klinis** Terdapat tiga segmen pasien yang memerlukan pendekatan medis berbeda:
   * **Kelompok Muda dengan Tekanan Darah Rendah** (Rerata: 25-27 tahun, TD: 47-62 mmHg):  
      Potensi gangguan metabolik dini, memerlukan pemantauan pemicu hipotensi.
   * **Dewasa dengan Tekanan Darah Normal-Rendah** (Rerata: 31-38 tahun, TD: 77-84 mmHg):  
      Manajemen kesehatan preventif untuk mempertahankan stabilitas.
   * **Lansia dengan Tekanan Darah Tinggi (Hanya Teridentifikasi di Excel)** (Rerata: 52 tahun, TD: 96 mmHg):  
      Kelompok prioritas untuk intervensi hipertensi dan skrining diabetes.
3. **Implikasi Kesehatan Masyarakat**
   * **Hipotensi Dominan**: 100% klaster di Python dan 66% di Excel masuk kategori hipotensi, mengindikasikan perlunya evaluasi penyebab tekanan darah rendah pada populasi ini.
   * **Stratifikasi Usia**: Klaster lansia (hanya terdeteksi di Excel) memverifikasi korelasi usia dengan peningkatan risiko hipertensi.
4. **Rekomendasi**
   * Penambahan variabel klinis (seperti kadar glukosa) untuk memperdalam analisis risiko diabetes.
   * Validasi temuan tekanan darah rendah dengan studi lanjutan terkait dampak jangka panjang.

**Penutup** Analisis ini berhasil memetakan keragaman profil pasien menggunakan pendekatan statistik yang solid. Temuan dapat menjadi dasar rekomendasi kebijakan kesehatan berbasis risiko spesifik kelompok.