# LAPORAN PROYEK DATA MINING

## *Segmentation of BPJS Health Insurance 2021 Data Using K-Means Clustering*


**Disusun oleh:**

**Kelompok 09**

| | |
|---|---|
| 12S20014 | Lidia Ginting |
| 12S20036 | Winda Sari Butarbutar |
| 12S20045 | Christine Hutagaol |

**PROGRAM STUDI SARJANA SISTEM INFORMASI**

**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO**

**INSTITUT TEKNOLOGI DEL**

**2023**

# DAFTAR ISI

# DAFTAR TABEL

# DAFTAR GAMBAR

# BAB 1
# BUSINESS UNDERSTANDING

Business understanding adalah tahap awal dari proses data mining yang bertujuan untuk memahami kebutuhan bisnis dan tujuan dari proyek data mining. Tahap ini penting untuk memastikan bahwa proyek data mining yang dilakukan sesuai dengan kebutuhan bisnis dan menghasilkan hasil yang bermanfaat.

## 1.1 *Determine Business Objective*

Analisis data BPJS Kesehatan dari rentang tahun 2015 hingga 2021 menjadi fokus utama, terutama pada data terkini tahun 2021. Tujuan utamanya adalah memahami pola biaya verifikasi dan bobot pada tahun terbaru yakni 2021 menggunakan teknik k-means clustering. Data sampel yang terdiri dari variabel biaya verifikasi (PNK18) dan bobot (PSTV15) menjadi dasar analisis ini.

Penggunaan teknik k-means clustering dipilih karena teknik ini sesuai dalam mengelompokkan data berdasarkan karakteristik yang ada, memungkinkan identifikasi pola kelompok-kelompok yang muncul berdasarkan biaya verifikasi dan bobot. Keunggulan k-means terletak pada kemampuannya menghadapi dataset yang cukup besar, ideal untuk analisis dataset BPJS Kesehatan yang melibatkan data dari rentang waktu yang luas.

Segmentasi yang dihasilkan dari analisis ini membantu dalam menunjukkan pola-pola dalam data. Dengan pemahaman yang lebih mendalam tentang pola ini, diharapkan dapat mendukung pengambilan keputusan yang lebih tepat. Melalui k-means, analisis ini berupaya untuk menemukan kelompok-kelompok dengan perbedaan signifikan dalam biaya verifikasi dan bobot, yang dapat memberikan wawasan berharga untuk keperluan bisnis dan operasional di bidang layanan kesehatan.

Adapun Business objective atau tujuan bisnis dari penggunaan Clustering (seperti K-Means atau Hierarchical Clustering) untuk mengelompokkan peserta BPJS berdasarkan pola biaya penggunaan layanan kesehatan sebagai berikut:

1. Segmentasi Pelanggan: Membuat segmentasi yang lebih baik terhadap pelanggan berdasarkan variabel biaya verifikasi dan bobot. Tujuannya bisa untuk memahami preferensi atau perilaku pelanggan, sehingga bisa meningkatkan pelayanan atau strategi pemasaran.
2. Optimalisasi Biaya: Menemukan pola dalam biaya verifikasi dan bobot untuk mengidentifikasi area-area di mana perusahaan bisa melakukan penghematan atau pengoptimalan biaya.
3. Perbaikan Layanan Kesehatan: Menyegmentasi pelayanan kesehatan (FKTP dan FKTRL) berdasarkan biaya verifikasi dan bobot untuk mengidentifikasi area-area yang memerlukan perbaikan atau peningkatan layanan.
4. Pengelompokan Efisiensi Operasional: Mengelompokkan unit-operasional berdasarkan biaya verifikasi dan bobot untuk meningkatkan efisiensi dan produktivitas.
5. Penentuan Kebutuhan Pasar: Mengidentifikasi pola konsumen atau pasar berdasarkan biaya verifikasi

dan bobot, membantu dalam pengembangan produk atau layanan yang lebih sesuai dengan kebutuhan.

## 1.2 *Technical Goal*

Tujuan dari proyek ini adalah mengembangkan sebuah model data mining utuk melakukan klastering pada sample BPJS data 2015-2021 dengan menggunakan algorithn K-Means. Tujuan utama dari analisis k-means pada data BPJS Kesehatan tahun 2021 adalah untuk memahami kelompok-kelompok yang muncul berdasarkan biaya verifikasi dan bobot. Dengan melakukan pengelompokan ini, dapat diperoleh pemahaman yang lebih dalam tentang pola-pola yang mungkin ada dalam data biaya verifikasi dan bobot pada tahun terbaru (tahun 2021). Melalui segmentasi ini, diharapkan dapat ditemukan kelompok-kelompok yang memiliki karakteristik atau pola yang berbeda-beda, yang kemudian dapat memberikan wawasan yang berguna untuk pengambilan keputusan yang lebih tepat, baik dalam optimalisasi biaya, evaluasi layanan kesehatan, maupun pemahaman yang lebih baik terhadap data BPJS Kesehatan secara keseluruhan.

## 1.3 *Produce Project Plan*

Dalam proyek ini, Python digunakan sebagai bahasa pemograman. Python adalah bahasa pemograman yang populer untuk Data Science, Machine Learning, dan Internet of Things (IoT). Python adalah bahasa pemograman yang digunakan untuk eksekusi sejumlah instruksi multiguna secara langsung dengan metode orientasi objek(OOP). Python juga menggunakan semantik dinamis meningkatkan keterbacaan kode.

Adapun algoritma yang digunakan adalah K-means, algoritma ini bertujuan untuk mengelompokkan data ke dalam kelompok yang homogen berdasarkan atribut tertentu, dengan upaya mencari pusat kelompok (centroids) yang optimal agar jarak antara setiap titik data ke centroid kelompoknya minimal. Algoritma k-means berupaya meminimalkan varians dalam setiap kelompok dengan menyesuaikan lokasi centroids, memungkinkan identifikasi pola atau struktur yang tersembunyi dalam data. Dengan pengelompokan yang mudah dipahami, k-means memberikan wawasan yang bermanfaat untuk pengambilan keputusan dalam berbagai bidang. Berikut adalah rencana proyek yang akan digunakan pada proyek ini.

*Table 1 Rencana Proyek*

| Aktivitas | Detail | Durasi |
|---|---|---|
| Pemilihan Kasus | Pemilihan Kasus | 1 Hari |
| | Pemilihan Algoritam | 1 Minggu |
| Business Understanding | Menentukan Objektif Bisnis | 2 Hari |
| | Menentukan Tujuan Bisnis | 3 Hari |
| | Membuat Rencana Proyek | 1 Hari |

| | | |
|---|---|---|
| Data Understanding | Mengumpulkan data | 3 Hari |
| | Menelaah Data | 4 Hari |
| | Memvalidasi Data | 2 Hari |
| Data Preparation | Memilah Data | 2 Hari |
| | Membersihkan Data | 2 Hari |
| | Mengkontruksi Data | 2 Hari |
| | Menentukan Label Data | 2 Hari |
| | Mengintegrasikan Data | 2 Hari |
| Modeling | Membangun Skenario Pengujian | 5 Hari |
| | Membangun Model | 3 Hari |
| Model Evaluation | Mengevaluasi Hasil Pemodelan | 4 Hari |
| | Melakukan Review Proses Pemodelan | 4 Hari |
| Deployment | Melakukan Deployment Model | 5 Hari |
| | Membuat Laporan akhir Proyek | 4 Hari |

# BAB 2
# DATA UNDERSTANDING

Data understanding merujuk pada proses menyeluruh untuk memahami dan mengeksplorasi data secara mendalam sebelum melakukan analisis lebih lanjut. Tahap ini melibatkan penelusuran yang cermat terhadap karakteristik data, termasuk jenis variabel yang ada, kualitas data, serta eksplorasi visual untuk mengidentifikasi pola, tren, dan distribusi data.

## 2.1 *Collect Initial Data*

Pengumpulan data merupakan langkah awal untuk menentukan data yang digunakan pada proyek, adapun dataset yang digunakan untuk mengelompokkan dan mengklasifikasikan data bpjs adalah “Data Sampel BPJS Kesehatan 2015-2021".

## 2.2 *Describe Data*

Dataset *Data Sampel BPJS Kesehatan 2015-2021* digunakan untuk mengelompokkan data peserta BPJS. Dataset ini terdiri dari 20 variabel (PSTV01, PSTV02, PSTV15, PNK02, PNK03, PNK04, PNK05, PNK06, PNK07, PNK08, PNK09, PNK10, PNK11, PNK12, PNK13, PNK13A, PNK14, PNK15, PNK16, PNK17, PNK18, PNK19, PNK20) dimana untuk melakukan clustering digunakan dua variabel yakni kedua variabel tersebut adalah variabel biaya verifikasi (PNK18) dan variabel bobot (PSTV15) yang memperlihatkan data. Untuk memperjelas data yang digunakan, dapat dilihat penjelasan pada tabel.

*Table 2 Variabel Data*

| No | Variabel | Label variabel | Deskripsi | Keterangan |
|---|---|---|---|---|
| 1 | PSTV01 | Nomor Peserta | Nomor identifikasi peserta yang bersifat unik dan telah dideidentifikasi untuk melindungi identitas peserta sebenarnya | Tidak Digunakan |
| 2 | PSTV02 | Nomor keluarga | Nomor yang mengidentifikasi kepala keluarga dalam sampel dan berfungsi sebagai penanda keluarga (peserta BPJS Kesehatan dalam satu keluarga memiliki nomor kepala keluarga yang sama) | Tidak Digunakan |
| 3 | PSTV15 | Bobot | Faktor pengali yang menggambarkan jumlah individu di dalam populasi diwakili oleh individu di dalam sampel | Digunakan |
| 4 | PNK02 | ID Kunjungan | Nomor identifikasi unik untuk menandakan setiap kunjungan FKTP oleh peserta | Tidak Digunakan |
| 5 | PNK03 | Tanggal kunjungan | Tanggal melakukan kunjungan | Tidak Digunakan |
| 6 | PNK04 | Tanggal tindakan | Tanggal melakukan tindakan | Tidak Digunakan |

| | | | | |
|---|---|---|---|---|
| 7 | PNK05 | Tanggal pulang | Tanggal menyelesaikan kunjungan | Tidak Digunakan |
| 8 | PNK06 | Provinsi faskes | Provinsi tempat peserta mengakses fasilitas kesehatan tingkat pratama (FKTP) | Tidak Digunakan |
| 9 | PNK07 | Kode Kab/Kota faskes | Kabupaten/kota tempat peserta mengakses fasilitas kesehatan tingkat pratama (FKTP) | Tidak Digunakan |
| 10 | PNK08 | Kepemilikan faskes | Kepemilikan dari fasilitas kesehatan tingkat pratama (FKTP) tempat peserta berkunjung | Tidak Digunakan |
| 11 | PNK09 | Jenis faskes | Jenis dari fasilitas kesehatan tingkat pertama (FKTP) tempat peserta berkunjung | Tidak Digunakan |
| 12 | PNK10 | Tipe faskes | Tipe dari fasilitas kesehatan tingkat pertama (FKTP) tempat peserta berkunjung | Tidak Digunakan |
| 13 | PNK11 | Tingkat layanan | Tingkat layanan yang diterima peserta di FKTP | Tidak Digunakan |
| 14 | PNK12 | Segmen peserta | Segmen peserta saat mengakses FKTP | Tidak Digunakan |
| 15 | PNK13 | Kode dan Nama diagnosis berdasarkan ICD-10 (3 digit) | Kode dan nama diagnosis berdasarkan 3 digit pertama kode ICD 10 yang diperoleh dari hasil input sistem informasi BPJS Kesehatan | Tidak Digunakan |
| 16 | PNK13A | Kode diagnosis berdasarkan ICD-10 (3 digit) | Kode dan nama diagnosis berdasarkan 3 digit pertama kode ICD 10 yang diperoleh dari hasil input sistem informasi BPJS Kesehatan | Tidak Digunakan |
| 17 | PNK14 | Kode diagnosis (3-5 digit) | Kode diagnosis menurut ICD 10 (jumlah digit tidak sama pada semua observasi dengan rentang 3-5 digit kode ICD 10) | Tidak Digunakan |
| 18 | PNK15 | Nama Diagnosis | Nama diagnosis yang terbaca oleh sistem informasi BPJS Kesehatan berdasarkan kode diagnosis yang ter-input dalam sistem | Tidak Digunakan |
| 19 | PNK16 | Nama Tindakan | Nama jenis tindakan yang dilakukan kepada pasien | Tidak Digunakan |
| 20 | PNK17 | Biaya tagih | Biaya yang ditagihkan fasilitas kesehatan untuk setiap ID kunjungan | Tidak Digunakan |
| 21 | PNK18 | Biaya verifikasi | Biaya yang diverifikasi BPJS Kesehatan untuk setiap nomor ID Kunjungan | Digunakan |
| 22 | PNK19 | Hasil pemeriksaan | Hasil pemeriksaan gula darah puasa (GDP) 1 | Tidak Digunakan |
| 23 | PNK20 | Hasil pemeriksaan | Hasil pemeriksaan gula darah puasa (GDP) 2 | Tidak Digunakan |

Data di atas merupakan data yang menunjukkan bahwa variabel yang digunakan adalah bobot perserta dan Biaya verifikasi peserta Diabetes Mellitus pada pelayanan FKTP Non Kapitasi yang terdaftar dalam survei di tahun 2021 yang dilakukan penulis *Data Sampel BPJS Kesehatan 2015-2021.*

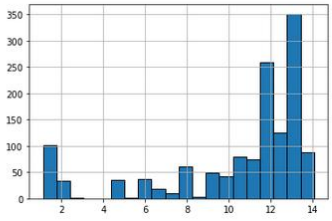
## 2.3 *Validation Data*

Tahap ini akan mencakup validasi data untuk memeriksa keakuratan, kelengkapan, dan kualitas data yang akan digunakan. Validasi bertujuan untuk mencegah terjadinya kesalahan atau masalah dalam data, seperti keberadaan nilai yang hilang. Proses validasi melibatkan pemeriksaan terhadap potensi gangguan atau ketidakakuratan dalam sumber data. Jika terdeteksi adanya gangguan, langkah selanjutnya adalah membersihkan data guna memastikan bahwa data yang digunakan konsisten, lengkap, dan akurat.

- Atribut PSTV15

```
In [21]: df['PSTV15'].describe()
Out[21]: count    1368.000000
         mean       10.438612
         std         3.634073
         min         1.115794
         25%         9.339209
         50%        11.736576
         75%        12.950229
         max        14.071786
         Name: PSTV15, dtype: float64

In [12]: df['PSTV15'].hist(edgecolor='black', bins=20);
```

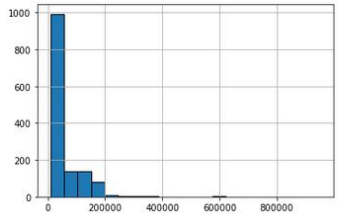



*Gambar 1 Validasi Data Atribut PSTV15*

- Atribut PNK18

```
In [13]: df['PNK18'].describe()
Out[13]: count      1368.000000
         mean      56556.673977
         std       56170.589770
         min       10000.000000
         25%       30000.000000
         50%       45000.000000
         75%       60000.000000
         max      950000.000000
         Name: PNK18, dtype: float64

In [14]: df['PNK18'].hist(edgecolor='black', bins=20);
```




*Gambar 2 Validasi Data Atribut PNK18*

# BAB 3

# DATA PREPARATION

Preprocessing adalah proses untuk mengubah data mentah menjadi data yang siap untuk dianalisis. Proses ini dilakukan untuk memperbaiki kualitas data, sehingga data dapat digunakan secara efektif dan efisien. Secara umum, proses preprocessing terdiri dari beberapa tahap, yaitu: Data cleaning, yaitu proses untuk memperbaiki data yang rusak atau tidak lengkap,Data transformation, yaitu proses untuk mengubah format data atau nilai data agar sesuai dengan kebutuhan analisis, Data reduction, yaitu proses untuk mengurangi ukuran data dengan cara menghilangkan data yang tidak penting atau data yang redundant, Data integration, yaitu proses untuk menggabungkan data dari berbagai sumber menjadi satu dataset, Proses preprocessing sangat penting untuk dilakukan, karena data mentah yang diperoleh dari berbagai sumber seringkali tidak sesuai dengan kebutuhan analisis. Data mentah dapat mengandung kesalahan, tidak lengkap, atau formatnya tidak sesuai. Proses preprocessing dapat membantu untuk memperbaiki kesalahan data, melengkapi data yang hilang, mengubah format data, dan mengurangi ukuran data.

Pada tahap ini, ada beberapa hal yang harus dilakukan terhadap dataset yang digunakan dalam membangun model, yaitu Memeriksa kelengkapan data, yaitu memastikan bahwa semua data yang diperlukan untuk analisis tersediaan, kemudian memeriksa akurasi data, yaitu memastikan bahwa data tidak mengandung kesalahan, lalu ada lagi mengubah format data, yaitu mengubah format data agar sesuai dengan kebutuhan analisis, kemudian menghapus data yang tidak relevan, yaitu menghapus data yang tidak penting atau data yang redundant, dan yang terakhir mengisi data yang hilang, yaitu mengisi data yang hilang dengan cara estimasi atau imputation. Proses preprocessing harus dilakukan dengan hati-hati, karena kesalahan pada proses ini dapat mempengaruhi hasil analisis.

## 3.1 *Data Cleaning*

Untuk mendapatkan hasil data yang berkualitas, perlu dilakukan pembersihan data. Pembersihan data dapat dilakukan dengan cara menghilangkan baris yang duplikat
Berikut merupakan gambar yang menunjukkan code untuk mengilangkan baris yang duplikat:

```
In [11]: # Menghilangkan Baris Duplikat
         df.drop_duplicates(inplace=True)

In [12]: df = df.dropna(axis=0)
```

*Gambar 3 Menghilangkan baris*

Selain menghilangkan baris yang duplikat, langkah data cleaning yang dapat dilakukan selanjutnya adalah menghapus baris atau kolom yang tidak memiliki nilai. Hal ini dilakukan untuk mempermudah pemrosesan data.

Berikut merupakan gambar yang menunjukkan code untuk menghapus baris atau kolom yang tidak memiliki nilai:

```
In [14]: # Menghapus Baris atau Kolom dengan Nilai yang Hilang
         df.dropna(axis=0)  # untuk menghapus baris
         df.dropna(axis=1)  # untuk menghapus kolom
```

*Gambar 4 Menghapus baris atau kolom*

Setelah itu akan dilakukan penyaringan data yang akan digunakan untuk membuat model. Penyaringan ini dilalukan untuk menghilangkan data yang tidak relevan atau tidak akurat.

Berikut merupakan gambar yang menunjukkan code untuk menyaring data :

```
In [15]: # Menyaring data hanya untuk tahun 2021
         dataframe_2020 = df[df['PNK05'] == 2021]

         # Menampilkan isi dataframe setelah penyaringan
         print(dataframe_2020)

         Empty DataFrame
         Columns: [PSTV01, PSTV02, PSTV15, PNK02, PNK03, PNK04, PNK05, PNK06, PNK07, PNK08, PNK09, PNK10, PNK11, PNK12, PNK13, PNK13A,
         PNK14, PNK15, PNK16, PNK17, PNK18, PNK19, PNK20]
         Index: []

         [0 rows x 23 columns]

In [16]: # Menghapus baris dengan tahun 2015 sampai 2019
         tahun_tidak_ingin = [2015, 2016, 2017, 2018, 2019, 2020]
         dataframe = df[~df['PNK05'].isin(tahun_tidak_ingin)]

         # Menampilkan isi dataframe setelah penghapusan
         print(dataframe)
```

*Gambar 5 Penyaringan data*

## 3.2 *Data Construction*

Pemilihan fungsi yang akan dipakai dalam membangun model merupakan hal yang penting untuk dilakukan dan berikut akan dijelaskan dan ditampilkan fitur yang akan digunakan. Pada proyek ini akan dilakukan pemilihan fitur antara lain ‘PSTV15’ dan ‘PNK18’. Berikut code yang digunakan untuk melakukan pemilihan fitur

```
In [29]: df= df[['PSTV15', 'PNK18']]
```

*Gambar 6 Pemilihan fitur*

Selanjutnya yang akan dilakukan adalah menghapus outlier yang bertujuan untuk meningkatkan hasil dengan signifikan

```
In [30]: df['PSTV15'].idxmax()
Out[30]: 814

In [31]: df.iloc[814]
Out[31]: PSTV15       14.071786
         PNK18     20000.000000
         Name: 814, dtype: float64
```

*Gambar 7 Menghapus outlier*

### 3.3 *Labeling Data*

Labelling data merupakan proses untuk memberikan informasi tambahan tentang penanda data, yang mewakili satu titik data atau nilai yang berasal dari sel dataset. Dimulai dengan proses pengumpulan data, melalui pembersihan dan transformasi data.

```
In [32]: X = df[['PSTV15']]
         y = df['PNK18']
         m = len(y)
```

*Gambar 8 Labeling data*

### 3.4 *Data Integration*

Mengintegrasikan data diterapkan untuk menggabungkan data dari berbagai sumber ke penyimpanan data yang koheren seperti gudang data(data warehouse). Sehingga tahapan ini tidak dilakukan karena data yang digunakan untuk membangun model berasal dari satu sumber dataset berformat dta yang di konversi kedalam bentuk csv.

# BAB 4
# MODELING

Pada bab ini akan menjelaskan Testing scenario dan modeling building, berikut merupakan penjelasannya.

## 4.1 *Testing Scenario*

Berikut langkah-langkah yang dilakukan meliputi :

***1.*** Data preparation

- ***a.*** Data cleaning
- ***b.*** Data construction
- ***c.*** Labelling Data

***2.*** Modelling

- ***a.*** Membangun Skenario
- ***b.*** Pengujian
- c. Membangun model

***3.*** Evaluation

- ***a.*** Membangun Hasil
- ***b.*** Pemodelan
- c. Melakukan Review Proses Pemodelan

Model kalsifikasi yang diharapkan berdasarkan yang dibangun memenuhi syarat berupa nilai dari SC > 55%

## 4.2 *Model Building*

Pada sub bab ini akan dijelaskan tahapan pada pembangunan model clustering k-means yang akan digunakan untuk mengelompokkan data pada data set. Berikut ini merupakan proses membangun model:

1. Melakukan import untuk Library yang akan digunakan pada algoritma Clustering. Berikut merupakan library yang akan digunakan:

```
In [19]: import pandas as pd
         import matplotlib.pyplot as plt
         import numpy as np
         from sklearn.cluster import KMeans
         from sklearn.metrics import silhouette_score
```

*Gambar 9 Import library*

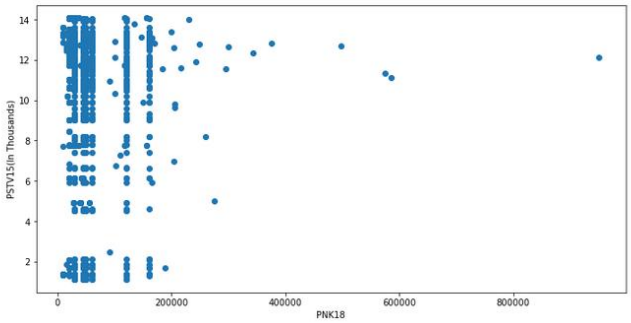
2. Membaca dataset yang akan digunakan untuk membangun model

```
In [32]: # Baca data
         data = pd.read_csv('DM2021_fktpnonkapitasi.csv')
         X = data[['PNK18', 'PSTV15']]
```

*Gambar 10 Membaca data*

3. Memvisualisasikan data dengan menggunakan scatter plot.

```
In [33]: #Visualise data points
         plt.figure(figsize=(12,6))
         plt.scatter(X['PNK18'], X["PSTV15"])
         plt.xlabel ('PNK18')
         plt.ylabel('PSTV15(In Thousands)')
         plt.show()
```




*Gambar 11 Memvisualisasikan data dengan scatter plot*

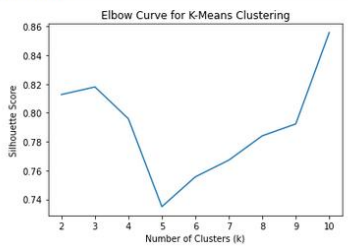
4. K-Means Clustering pada dataset yang telah dibaca dan kemudian menganalisis nilai Silhouette Coefficient untuk evaluasi clustering.

```
In [34]: # Calculate silhouette scores for a range of k values
         k_range = range(2, 11)
         silhouette_scores = []
         for k in k_range:
             kmeans = KMeans(n_clusters=k)
             kmeans.fit(data[["PNK18"]])
             clusters = kmeans.labels_
             silhouette_scores.append(silhouette_score(data[["PNK18"]], clusters))
```

*Gambar 12 Silhouette Coefficient untuk evaluasi clustering*

5. Hasil silhouette score untuk setiap nilai k disimpan dalam sebuah list (silhouette_scores). Setelah iterasi selesai, code tersebut memplot hasilnya dalam bentuk kurva menggunakan matplotlib. Pada sumbu x, ditampilkan jumlah kluster (k), dan pada sumbu y, ditampilkan nilai silhouette score. Tujuannya adalah untuk melihat "elbow" (siku) dalam kurva, yang menunjukkan di mana penambahan jumlah kluster tidak lagi memberikan peningkatan signifikan dalam nilai silhouette score, dan ini dapat dianggap sebagai jumlah kluster yang optimal untuk data yang diberikan.

```
In [35]: # Plot the elbow curve
         plt.plot(k_range, silhouette_scores)
         plt.xlabel("Number of Clusters (k)")
         plt.ylabel("Silhouette Score")
         plt.title("Elbow Curve for K-Means Clustering")
         plt.show()
```




*Gambar 13 Hasil dalam bentuk kurva*

6. Pada code berikut kita akan menentukan jumlah klaster optimal(k), dan melakukan k-means clustering menggunakan algoritma k-means.

```
In [36]: # Choose the optimal number of clusters (k)
k = 3

# Perform K-Means clustering
kmeans = KMeans(n_clusters=k)
kmeans.fit(data[["PNK18"]])
clusters = kmeans.labels_

# Add cluster labels to the data
data["Cluster"] = clusters

# Mendapatkan kode PNK18 yang paling umum untuk setiap klaster
cluster_codes = []
for i in range(k):
    cluster_data = data[data["Cluster"] == i]
    # Gunakan `dropna()` untuk menghilangkan baris yang memiliki value NaN untuk atribut PNK18
    cluster_data = cluster_data.dropna(subset=["PNK18"])
    cluster_codes.append(cluster_data["PNK18"].value_counts().nlargest(10).index.tolist())

# Mencetak kode PNK18 yang paling umum untuk setiap klaster
for i, codes in enumerate(cluster_codes):
    print(f"Klaster {i + 1} PNK18 yang paling umum:", codes)

Klaster 1 PNK18 yang paling umum: [30000, 45000, 20000, 60000, 50000, 10000, 28000, 15000, 40000, 55000]
Klaster 2 PNK18 yang paling umum: [120000, 160000, 118000, 156000, 100000, 165000, 91000, 204000, 102000, 249400]
Klaster 3 PNK18 yang paling umum: [498000, 585000, 950000, 575000]
```

*Gambar 14 Jumlah klaster optimal(k)*

7. Selanjutnya kita akan menampilkan centroid

```
In [38]: # Menampilkan centroid
centroids = kmeans.cluster_centers_
print("Posisi Centroid:")
print(centroids)

Posisi Centroid:
[[3.62459964e+04 1.04599777e+01]
 [1.41754292e+05 1.03154371e+01]
 [6.52000000e+05 1.18251875e+01]]
```

*Gambar 15 Tampilan centroid*

8. `kmeans.inertia_` adalah atribut yang mengembalikan nilai inersia dari model KMeans setelah dilatih pada data. Inersia dalam konteks KMeans mengacu pada jumlah kuadrat jarak antara setiap sampel dan centroid terdekatnya di dalam klasternya.Secara matematis, inersia dihitung sebagai jumlah kuadrat jarak Euclidean antara setiap sampel dengan centroid klasternya dan kemudian dijumlahkan untuk semua sampel dalam klaster

```
In [26]: print(kmeans.inertia_)

689110013463.5686
```

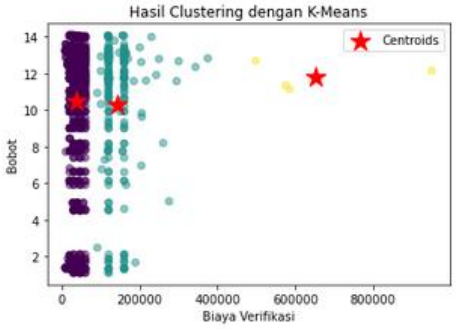
*Gambar 16 Pemanggilan kmeans.inertia_*

9. Langkah selanjutnya adalah melatih model k-means pada data untuk melakukan pengelompokan data kedalam klaster berdasarkan pola yang ada didalamnya setelah itu pada data akan ditambahkan label untuk memungkinkan melihat dan menganalisis data mana yang termasuk kedalam klaster yang mana. Setelah dilakukan pelatihan terhadap model dan menambahkan label pada klaster, akan dilakukan visualisasi data terhadap hasil clustering dan centroids.

In [46]:
```
# Latih model KMeans pada data
kmeans.fit(features)

# Menambahkan label cluster ke data
data['Cluster'] = kmeans.labels_

# Visualisasi hasil clustering dan centroids
plt.scatter(data['PNK18'], data['PSTV15'], c=data['Cluster'], cmap='viridis', alpha=0.5)
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], marker='*', s=300, c='red', label='Centroids')
plt.xlabel('Biaya Verifikasi')
plt.ylabel('Bobot')
plt.title(f'Hasil Clustering dengan K-Means')
plt.legend()
plt.show()
```




*Gambar 17 Melatih modek K-Means*

# BAB 5
# MODEL EVALUATION

## 5.1 *Result Evaluation*

Pada sub bab ini akan ditampilkan hasil evaluasi dari model yang dibangun. Adapun metrik evaluasi dari model yang dibangun adalah Silhouette Coefficient. Dalam evaluasi pengelompokan data, Koefisien Silhouette menjadi salah satu metrik penting yang digunakan untuk mengevaluasi seberapa baik pengelompokan telah dilakukan. Berikut merupakan code yang menunjukkan hasil evaluasi:

```
In [30]: # Menghitung nilai Silhouette Coefficient
         silhouette_avg = silhouette_score(features, kmeans.labels_)

         # Konversi nilai SC ke dalam persen
         silhouette_avg_percent = silhouette_avg * 100

         print(f'Nilai Silhouette Coefficient: {silhouette_avg_percent:.2f}%')

         Nilai Silhouette Coefficient: 81.79%
```

*Gambar 18 Menghitung nilai  Silhouette Coefficient*

## *5.2 Result Review*

Pada sub bab ini akan dibahas mengenai review hasil yang didapatkan dari hasil evaluasi.

Adapun syarat yang harus dipenuhi pada evaluasi clustering adalah : SC>55%

Sedangkan hasil evaluasi dari model klastering yang dibangun adalah 81,97% yang berarti hasil evaluasi dari model memenuhi persyaratan yang ditentukan.

# BAB 6
# DEPLOYMENT

## 6.1 *Deployment*

Setelah tahap modelling telah dilakukan dan evaluasi terhadap model sudah dilakukan. Kemudian model akan disimpan dalam library python yaitu dalam bentuk file pickel yang membuat model dapat digunakan dengan mudah terhadap sistem yang akan di deploy

```
In [48]: # Simpan model KMeans dan nilai Silhouette Coefficient ke dalam file
         with open('kmeans_model.pkl', 'wb') as model_file:
             pickle.dump(kmeans, model_file)

         with open('silhouette_coefficient.pkl', 'wb') as coef_file:
             pickle.dump(silhouette_avg_percent, coef_file)
```

*Gambar 19 Menyimpan model KMeans dan nilai Silhouette*

Pickel merupakan library yang digunakan untuk menyimpan dan membaca data kedalam atau dari suatu file berformat.pkl.

## *6.2 Web Application*

Pada tahap ini dilakukan persiapan sebelum interface dalam mengimplementasikan model, kerangka desain pada HTML juga akan didesain pada tahap ini. Tahap persiapan ini diperlukan agar dalam proses implementasi interface dengan HTML dilakukan dengan terstruktur.

Berikut merupakan folder yang menjadi lokasi dari design sistem.


*Gambar 20 Lokasi folder*

Beriku merupakan isi dari file HTML, CSS serta gambar yang digunakan untuk mendesaign sistem.


*Gambar 21 Isi folder*

### *6.2.1 Interface*

Dalam membuat sistem BPJS Participants clustering with K-Means Algorithm, pertama sekali dibuat desain HTML seperti berikut

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Sistem</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <div class="container">
      <div class="header-content">
        <img src="BPJS.png" alt="Logo" class="logo">
        <h1>BPJS participants clustering with K-Means Algorithm</h1>
        <nav>
          <a href="#">Home</a>
        </nav>
      </div>
    </div>
  </header>
  <main>
    <section class="left-column">
    <section class="form-section">
      <form>
        <h2>Isi form berikut</h2>

        <label for="">Provinsi:</label>
        <select id="provinsi" name="provinsi">
          <option value="aceh">Aceh</option>
          <option value="Sumatera Utara">Sumatera Utara</option>
          <option value="sumatera selatan">Sumatera Selatan</option>
          <option value="Sumatera Barat">Sumatera Barat</option>
          <option value="bengkulu">Bengkulu</option>
```

```html
          <option value="bengkulu">Bengkulu</option>
          <option value="lampung">Lampung</option>
          <option value="kalimantan barat">Kalimantan Barat</option>
          <option value="kalimantan timur">Kalimantan Timur</option>
          <option value="kalimantan selatan">Kalimantan Selatan</option>
          <option value="kalimantan tengah">Kalimantan Tengah</option>
          <option value="kalimantan utara">Kalimantan Utara</option>
          <option value="banten">Banten</option>
          <option value="DKI Jakarta">DKI Jakarta</option>
          <option value="jawa barat">Jawa Barat</option>
          <option value="jawa tengah">Jawa Tengah</option>
          <option value="jawa timur">Jawa Timur</option>
          <option value="bali">Bali</option>
          <option value="NTT">NTT</option>
          <option value="NTB">NTB</option>
          <option value="gorontalo">Gorontalo</option>
          <option value="sulawei selatan">Sulawesi Selatan</option>
          <option value="sulawesi tengah">Sulawesi Tengah</option>
          <option value="sulawesi barat">Sulawesi Barat</option>
          <option value="sulawei tenggara">Sulawesi Tenggara</option>
          <option value="papua tengah">Papua Tengah</option>
          <option value="papua pegunungan">Papua Pegunungan</option>
          <option value="papua barat daya">Papua Barat Daya</option>
        </select>


        <label for="nama">Kota:</label>
        <input type="text" id="kota" name="kota" required>
```

```html
        <label for="nama">Biaya:</label>
        <input type="text" id="biaya" name="biaya" required>

        <label for="email">Bobot:</label>
        <input type="email" id="bobot" name="bobot" required>

        <button type="submit">Hasil</button>
      </form>
    </section>
  </section>
<section class="mt-8 h-full flex-auto rounded-lg md:mt-0 md:ml-8">
  <div class="bg-white p-16 rounded-lg">
    <h1 class="text-2xl mb-8">
      Hasil Pengelompokan :
    </h1>
    <h2 class="text-5xl font-bold">{{ hasil }}</h2>

  </div>

</section>
  </main>
```

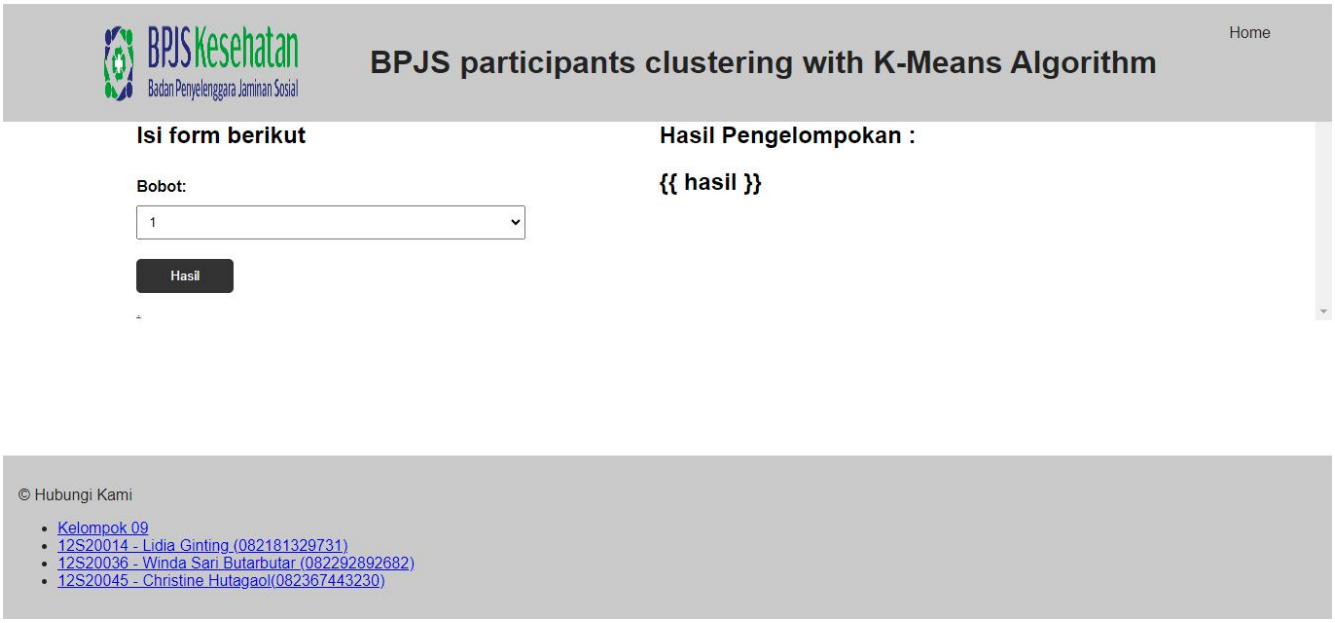
```html
  <footer>
    <div class="footer-content">
      <div class="footer-left">
        <p>&copy; Hubungi Kami</p>
      </div>
      <div class="footer-right">
        <ul>
          <li><a href="#">Kelompok 09</a></li>
          <li><a href="#">12S20014 - Lidia Ginting (08)</a></li>
          <li><a href="#">12S20036 - Winda Sari Butarbutar (08)</a></li>
          <li><a href="#">12S20045 - Christine Hutagaol(082367443230)</a></li>
        </ul>
      </div>
    </div>
  </footer>

</body>
</html>

```

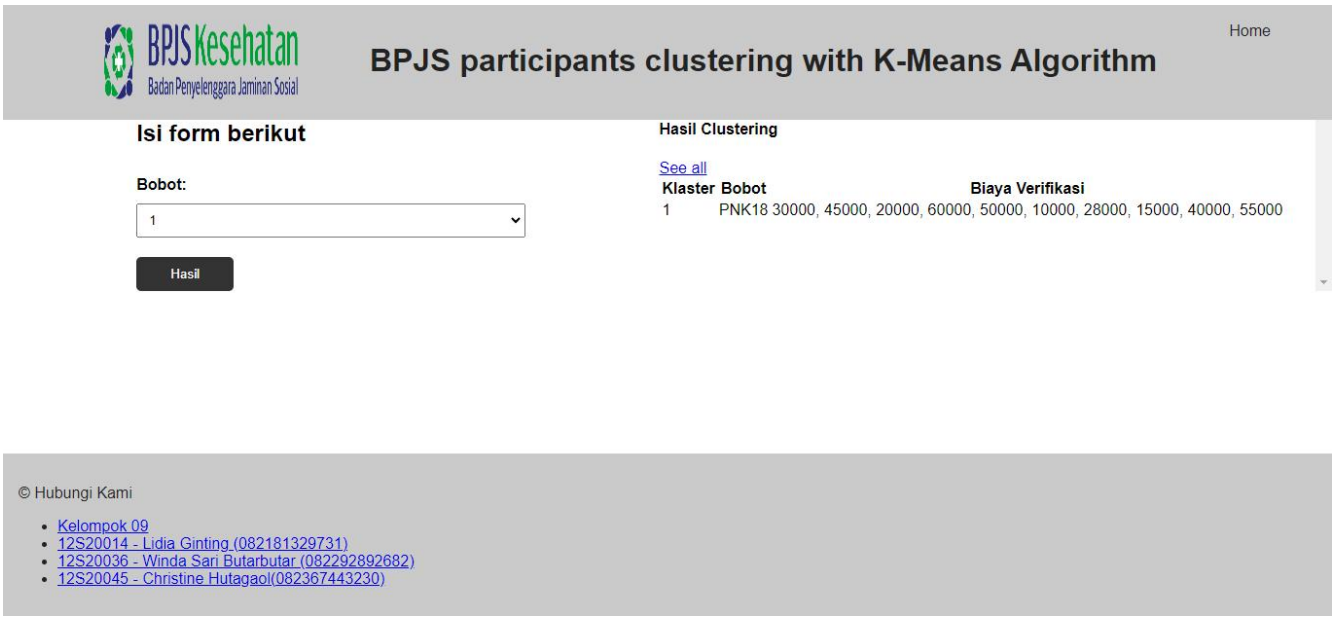
*Gambar 22 Code HTML sistem*

Sehingga menghasilkan interface sebagai berikut:




*Gambar 23 Sistem*

Dan berikut contoh hasil clustering data pada sistem




*Gambar 24 Hasil clustering data*