

DOKUMEN PROYEK

12S4054 - PENAMBANGAN DATA

Strategi Intervensi untuk Peserta BPJS Tuberkulosis Berdasarkan Segmentasi Usia dan Perilaku Akses Fasilitas Kesehatan Menggunakan K-Means Clustering



Disusun Oleh:

12S22007	Tamara Y Sianipar
12S22018	Jesica A Siburian
12S22021	Krisnia Calysta Siahaan
12S22042	Ruth Septiana Simanullang

**PROGRAM STUDI SARJANA SISTEM INFORMASI
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO (FITE)
INSTITUT TEKNOLOGI DEL
(2024/2025)**

DAFTAR ISI

DAFTAR ISI	2
BAB I BUSINESS UNDERSTANDING	4
1.1 Determine Business Objective.....	4
1.2 Determine Project Goal	4
1.3 Produce Project Plan	4
BAB II DATA UNDERSTANDING	6
2.1 Pengumpulan Data.....	7
2.1.1 Data Kontekstual TB	7
2.2 Menelaah Data	13
2.3 Memvalidasi Data	17
2.3.1 Pemeriksaan Data	17
2.3.2 Validasi Format dan Tipe Data	20
2.3.3 Korelasi Heatmap	24
BAB III DATA PREPARATION	29
3.1 Data Selection	29
3.1.1 Sumber Data	29
3.1.2 Pemilihan Variabel	29
3.2 Data Cleaning.....	31
3.2.1 Menghapus Nilai yang Hilang.....	31
3.2.2 Menghapus Duplikat	34
3.2.3 Memperbaiki Format Data	34
3.2.4 Menangani Kolom Kategorikal	35
3.2.5 Menyaring Kolom yang Tidak Relevan	36
3.3 Data Construct	37
3.3.1 Menghitung Usia Peserta.....	37
3.3.2 Menambahkan Fitur Berdasarkan Waktu (Bulan dan Tahun Kunjungan) FKRTL dan FKTP Non-Kapitasi	37
3.3.3 Menghitung Frekuensi Kunjungan	38
3.3.4 Segmentasi Berdasarkan Usia.....	38
3.3.5 Menambahkan Fitur Berdasarkan Fasilitas Kesehatan.....	38
3.4 Labeling Data	40

3.4.1 Menentukan Label Berdasarkan Usia	40
3.5 Data Integration	40
3.5.1 Menggabungkan Data Berdasarkan ID Peserta (PSTV01)	40
3.5.2 Memverifikasi Data Gabungan.....	41
BAB IV MODELLING.....	43
4.1 Membuat Model	43
4.1.1 Memilih Fitur yang Relevan	43
4.1.2 Melakukan Normalisasi Data	43
4.1.3 Menentukan Jumlah Klaster yang Optimal dengan Metode Elbow	43
4.1.4 Membangun Model K-Means dengan Jumlah Klaster Optimal	44
4.1.5 Visualisasi	45
BAB V EVALUATION	47
5.1 Evaluasi Hasil Klasterisasi	47
5.2 Evaluasi Visualisasi.....	47
5.3 Relevansi Klaster dengan Kebijakan Kesehatan	47
BAB VI DEPLOYMENT	49
6.1 Struktur dan Aplikasi Komponen Web	49
6.2 Langkah-Langkah Pembuatan Website	49
6.3 Kode Aplikasi Web	49
6.4 Langkah-langkah Deploy Aplikasi Web	53
6.5 Evaluasi Website	54

BAB I BUSINESS UNDERSTANDING

1.1 Determine Business Objective

BPJS Kesehatan merupakan lembaga yang bertanggung jawab dalam menjamin akses layanan kesehatan yang merata dan berkualitas bagi seluruh penduduk Indonesia. Seiring dengan meningkatnya jumlah peserta setiap tahun, tantangan dalam pengelolaan layanan kesehatan pun semakin kompleks, terutama untuk penyakit yang membutuhkan penanganan jangka panjang seperti Tuberkulosis (TB).

Tuberkulosis merupakan penyakit menular yang masih menjadi masalah kesehatan utama di Indonesia. Penanganan TB membutuhkan strategi layanan yang terstruktur dan alokasi sumber daya yang efektif, termasuk penyediaan fasilitas kesehatan, tenaga medis, serta edukasi masyarakat.

Untuk menjawab tantangan ini, BPJS Kesehatan memerlukan pendekatan berbasis data dalam memahami distribusi peserta yang menderita TB. Dengan informasi ini, BPJS Kesehatan dapat menyusun strategi intervensi yang lebih efektif, tepat sasaran, dan efisien dalam penggunaan sumber daya.

Tujuan bisnis utama dari proyek ini adalah:

- Meningkatkan efisiensi dan ketepatan sasaran dalam pengambilan keputusan strategis terkait penanganan Tuberkulosis.
- Mendukung perencanaan distribusi pelayanan, edukasi dan intervensi berbasis klaster yang lebih relevan.

1.2 Determine Project Goal

Agar tujuan bisnis dapat tercapai, proyek ini menetapkan beberapa sasaran teknis dan analitis sebagai berikut:

- Menggunakan teknik *K-Means Clustering* untuk mengelompokkan peserta BPJS berdasarkan dua faktor utama: usia dan perilaku mereka dalam mengakses fasilitas kesehatan.
- Merancang strategi intervensi yang disesuaikan dengan setiap kelompok usia dan pola perilaku akses fasilitas kesehatan.

1.3 Produce Project Plan

Untuk memastikan pelaksanaan proyek berjalan secara terstruktur, berikut adalah tahapan utama proyek beserta kebutuhan masing-masing:

A. Business Understanding

Pada tahap ini, aktivitas utama yang dilakukan adalah menentukan tujuan bisnis dan analisis kebutuhan kebijakan TB berbasis data. Kebutuhan yang digunakan di tahap ini adalah melakukan studi regulasi BPJS dan TB.

B. Data Understanding

Pada tahap ini, aktivitas utama yang dilakukan adalah melakukan eksplorasi metadata dan variabel Tuberkulosis dari file BPJS. Selain itu, melakukan identifikasi karakteristik peserta & pelayanan Tuberkulosis. Adapun kebutuhan yang digunakan dalam tahap ini adalah Data Sampel Final 2022 : Kontekstual TB.

C. Data Preparation

Pada tahap ini , aktivitas utama yang dilakukan adalah melakukan cleaning data (untuk data null dan data duplikat), feature engineering, dan normalisasi variabel numerik.

D. Modeling

Pada tahap ini , aktivitas utama yang dilakukan adalah menentukan jumlah kluster optimal, melakukan cluster menggunakan K-Means, Interpretasi hasil kluster. Kebutuhan yang diperlukan adalah data yang siap untuk dilakukan modeling.

E. Evaluation

Pada tahap ini, aktivitas utama yang dilakukan adalah melakukan evaluasi performa kluster (apakah dapat dibedakan dengan jelas?), hingga validasi domain (apakah kluster masuk akal bagi pengambil kebijakan atau keputusan?). Kebutuhan yang diperlukan untuk tahap ini adalah domain knowledge, metrik evaluasi internal.

F. Deployment

Pada tahap ini, aktivitas yang dilakukan adalah menyusun laporan akhir, visualisasi hasil segmentasi, hingga rekomendasi strategi intervensi tiap karakter.

BAB II DATA UNDERSTANDING

Bertujuan untuk memahami struktur dan karakteristik data BPJS 2015-2021 yang akan digunakan dalam proses klasifikasi, prediksi, dan/atau klastering. Terdapat lima subset data untuk data sampel BPJS Kesehatan 2015-2021, yaitu:

- 1. Kepesertaan**

Kepesertaan data sampel BPJS Kesehatan tahun 2015-2021 sebesar 2.305.435 peserta. Variabel pada data kepesertaan terdiri dari karakteristik peserta seperti, tanggal lahir, jenis kelamin, status perkawinan, hubungan keluarga dan tempat tinggal peserta (provinsi dan kabupaten/kota). Variabel 'Nomor Peserta' (PSTV01) adalah nomor unik dari seluruh peserta JKN-KIS yang sudah diidentifikasi untuk menjaga kerahasiaan identitas peserta.

- 2. Pelayanan Fasilitas Kesehatan Tingkat Pertama (FKTP) Kapitasi**

Total pelayanan FKTP sebesar 2.498.805 kunjungan.

- 3. Pelayanan Fasilitas Kesehatan Rujukan Tingkat Lanjut (FKRTL)**

Total pelayanan FKRTL sebesar 872.201 kunjungan

- 4. Pelayanan Fasilitas Kesehatan Tingkat Pertama (FKTP) Non Kapitasi**

Total pelayanan FKTP Non kapitasi tahun 2021 sebesar 95.617 tindakan

- 5. Pelayanan Fasilitas Kesehatan Rujukan Tingkat Lanjut (FKRTL) Diagnosis Sekunder**

Data diagnosis sekunder sebagai bagian dari data pelayanan FKRTL dengan total observasi 925.803 diagnosis.

2.1 Pengumpulan Data

2.1.1 Data Kontekstual TB

Data sampel kontekstual Tuberkulosis (TB) adalah data kepesertaan dan pelayanan dari peserta JKN-KIS melalui data warehouse BPJS Kesehatan. Peserta yang terpilih dalam data sampel kontekstual TB adalah mereka yang diidentifikasi pernah mendapatkan pelayanan FKTP maupun FKRTL dengan diagnosis TB melalui kode ICD-10 yaitu A15, A16, A17, A18 dan A19 pada tahun 2019.

1. TB20152021_fkrtl.dta

```
In [6]: df = pd.read_stata(r"C:\Users\Septiana\Downloads\Final Project\Data Sampel Final 2022\Kontekstual TB\TB20152021_fkrtl.dta")
# Tampilkan 5 baris pertama
df.head()
```

Out[6]:

	PSTV01	PSTV02	PSTV15	FKP02	FKL02	FKL03	FKL04	FKL05	FKL06	FKL07	...	FKL39	FKL40	FKL41	FKL42	FKL43	FKL44
0	96934726	96934726	0.899331	1000801150000582	2015-01-21	2015-01-21	PAPUA	JAYAPURA	Pemerintah kab/kota	...		NaN				NaN	
1	11315168	11315168	0.938273	1000801150001076	2015-01-13	2015-01-13	PAPUA	JAYAPURA	Pemerintah kab/kota	...		NaN				NaN	
2	48014275	20325942	0.998232	1000801160000002	2016-01-02	2016-01-06	PAPUA	JAYAPURA	Pemerintah kab/kota	...		0.0				0.0	
3	78552717	3459550	1.205653	1000801160000219	2016-01-08	2016-01-08	PAPUA	JAYAPURA	Pemerintah kab/kota	...		0.0				0.0	
4	48014275	20325942	0.998232	1000801160000321	2016-01-11	2016-01-11	PAPUA	JAYAPURA	Pemerintah kab/kota	...		0.0				0.0	

5 rows x 55 columns

Pelayanan FKRTL adalah data pelayanan dari data sampel peserta didiagnosis TB yang melakukan kunjungan ke pelayanan FKRTL pada tahun 2015-2021 sebesar 1.583.242 kunjungan.

Berikut ini detail data:

```
In [4]: df = pd.read_stata(r"C:\Users\Septiana\Downloads\Final Project\Data Sampel Final 2022\Kontekstual TB\TB20152021_fkrtl.dta")
df.info()
```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1583242 entries, 0 to 1583241
Data columns (total 55 columns):
Column Non-Null Count Dtype

0 PSTV01 1583242 non-null int32
1 PSTV02 1583242 non-null int32
2 PSTV15 1583242 non-null float32
3 FKP02 1583242 non-null object
4 FKL02 1583242 non-null object
5 FKL03 1583242 non-null datetime64[ns]
6 FKL04 1583242 non-null datetime64[ns]
7 FKL05 1583242 non-null category
8 FKL06 1583242 non-null category
9 FKL07 1583242 non-null category
10 FKL08 1583242 non-null category
11 FKL09 1583242 non-null category
12 FKL10 1583242 non-null category
13 FKL11 1583242 non-null category

Terdapat beberapa kolom/atribut yang tersedia di dalam data FKRTL ini, diantaranya:

1. PSTV01: Nomor Peserta
2. PSTV02: Nomor Keluarga
3. PSTV15: Bobot
4. FKP02: No Asal Rujukan
5. FKL02: ID Kunjungan

6. FKL03: Tanggal datang kunjungan FKRTL
7. FKL04: Tanggal pulang kunjungan FKRTL
8. FKL05: Provinsi FKRTL
9. FKL06: Kabupaten/Kota FKRTL
10. FKL07: Kepemilikan FKRTL
11. FKL08: Jenis FKRTL
12. FKL09: Tipe FKRTL
13. FKL10: Tingkat Pelayanan FKRTL
14. FKL11: Jenis Poli FKRTL
15. FKL12: Segmen Peserta saat akses layanan FKRTL
16. FKL13: Kelas iuran premi peserta saat akses layanan FKRTL
17. FKL14: Status pulang dari FKRTL
18. FKL15: Kode dan nama diagnosis masuk ICD 10
19. FKL15A: Kode diagnosis masuk ICD 10
20. FKL16: Kode ICD 10 diagnosis masuk FKRTL
21. FKL16A: Nama diagnosis masuk FKRTL
22. FKL17: Kode dan nama diagnosis primer ICD 10
23. FKL17A: Kode diagnosis primer ICD 10
24. FKL18: Kode ICD 10 diagnosis primer FKRTL
25. FKL18A: Nama diagnosis primer FKRLT
26. FKL19: Kode INACBGs
27. FKL19A: Deskripsi kode INACBGs
28. FKL20: INACBGs - Kode Casemix main groups (digit ke-1)
29. FKL21: INACBGs - Tipe kelompok kasus atau case groups (digit ke-2)
30. FKL22: INACBGs - Spesifikasi kelompok kasus (digit ke-3)
31. FKL23: INACBGs -Tingkat keparahan kelompok kasus (digit ke-4)
32. FKL25: Provinsi faskes perujuk
33. FKL26: Kabupaten/Kota faskes perujuk
34. FKL27: Kepemilikan faskes perujuk
35. FKL28: Jenis faskes perujuk
36. FKL29: Tipe faskes perujuk
37. FKL30: Jenis prosedur
38. FKL31: Tarif regional INACBGs
39. FKL32: Group Tarif INACBGs
40. FKL33: Kode special sub-acute groups (SA)
41. FKL34: Tarif special sub-acute groups (SA)
42. FKL35: Kode special procedures (SP)
43. FKL36: Deskripsi special procedures (SP)
44. FKL37: Tarif special procedures (SP)
45. FKL38: Kode special prosthesis (RR)

46. FKL39: Deskripsi special prosthesis (RR)
47. FKL40: Tarif special prosthesis (RR)
48. FKL41: Kode special investigation (SI)
49. FKL42: Deskripsi special investigation (SI)
50. FKL43: Tarif special investigation (SI)
51. FKL44: Kode special drugs (SD)
52. FKL45: Deskripsi special drugs (SD)
53. FKL46: Tarif special drugs (SD)
54. FKL47: Biaya Tagih - oleh fasilitas kesehatan (provider)
55. FKL48: Biaya verifikasi - BPJS Kesehatan setelah dilakukan verifikasi.

2. TB20152021_fkrtldxsekunder.dta

```
In [7]: df = pd.read_stata(r"C:\Users\Septiana\Downloads\Final Project\Data Sampel Final 2022\Kontekstual TB\TB20152021_fkrtldxsekunder.dta")
# Tampilkan 5 baris pertama
df.head()
```

	FKL02	FKL24	FKL24A	FKL24B
0	2603412140001163	H814	H81	H81 Disorders of vestibular function
1	1337801150000108	M199	M19	M19 Other arthrosis
2	1571901150000730	M4909	M49	M49 Spondylopathies in diseases classified els...
3	3039201150000052	E790	E79	E79 Disorders of purine and pyrimidine metabolism
4	3376701150000072	N390	N39	N39 Other disorders of urinary system

```
In [8]: df = pd.read_stata(r"C:\Users\Septiana\Downloads\Final Project\Data Sampel Final 2022\Kontekstual TB\TB2019_kepesertaan.dta")
```

3. TB2019_kepesertaan.dta

```
In [8]: df = pd.read_stata(r"C:\Users\Septiana\Downloads\Final Project\Data Sampel Final 2022\Kontekstual TB\TB2019_kepesertaan.dta")
# Tampilkan 5 baris pertama
df.head()
```

	PSTV01	PSTV02	PSTV03	PSTV04	PSTV05	PSTV06	PSTV07	PSTV08	PSTV09	PSTV10	PSTV11	PSTV12	PSTV13	PS'
0	21611150	21611150	1957-09-12	PESERTA	PEREMPUAN	KAWIN	KELAS I	PPU	ACEH	ACEH BESAR	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	A BE
1	94343049	96772919	1961-12-03	SUAMI	LAKI-LAKI	KAWIN	KELAS I	PPU	ACEH	ACEH TENGGARA	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	A TENGG
2	83393824	298548714	2002-10-05	ANAK	LAKI-LAKI	BELUM KAWIN	KELAS I	PPU	ACEH	ACEH TENGGARA	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	A TENGG
3	328537885	328537885	1989-07-13	PESERTA	PEREMPUAN	BELUM KAWIN	KELAS III	PBI APBD	ACEH	ACEH TAMIANG	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	A TAMI
4	67805935	67805935	1972-11-13	PESERTA	LAKI-LAKI	KAWIN	KELAS I	PPU	ACEH	ACEH TAMIANG	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	A TAMI

Data diatas merupakan data peserta BPJS Kesehatan tahun 2019 dengan diagnosa penyakit Tuberkulosis. Terdapat beberapa kolom/atribut yang tersedia di dalam data kepesertaan ini, diantaranya:

1. PSTV01: Nomor Peserta
2. PSTV02: Nomor Keluarga
3. PSTV03: Tanggal Lahir Peserta
4. PSTV04: Hubungan Keluarga
5. PSTV05: Jenis Kelamin
6. PSTV06: Status Perkawinan

7. PSTV07: Kelas Rawat
8. PSTV08: Segmentasi Peserta
9. PSTV09: Provinsi Tempat Tinggal Peserta
10. PSTV10: Kabupaten/Kota Tempat Tinggal Peserta
11. PSTV11: Kepemilikan Faskes
12. PSTV12: Jenis Faskes
13. PSTV13: Provinsi Fasilitas Kesehatan Peserta Terdaftar
14. PSTV14: Kabupaten/Kota Fasilitas Kesehatan Peserta Terdaftar
15. PSTV15: Bobot
16. PSTV16: Tahun Sampel
17. PSTV17: Status Kepesertaan
18. PSTV18: Tahun Meninggal

4. TB2020_kepesertaan.dta: Kepesertaan pasien TB tahun 2020

In [9]: `df = pd.read_stata(r"C:\Users\Septiana\Downloads\Final Project\Data Sampel Final 2022\Kontekstual TB\TB2020_kepesertaan.dta")`
`# Tampilkan 5 baris pertama`
`df.head()`

Out[9]:

	PSTV01	PSTV02	PSTV03	PSTV04	PSTV05	PSTV06	PSTV07	PSTV08	PSTV09	PSTV10	PSTV11	PSTV12	PSTV13
0	32853965	31945523	1958-11-27	SUAMI	LAKI-LAKI	KAWIN	KELAS I	PPU	ACEH	ACEH TENGAH	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH
1	8555967	8555967	1936-07-01	PESERTA	LAKI-LAKI	KAWIN	KELAS I	BUKAN PEKERJA	ACEH	ACEH TENGAH	POLRI	KLINIK PRATAMA	ACEH
2	8797049	249479	1950-07-21	SUAMI	LAKI-LAKI	KAWIN	KELAS I	BUKAN PEKERJA	ACEH	ACEH SELATAN	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH
3	83472658	54408870	2012-08-12	ANAK	PEREMPUAN	BELUM KAWIN	KELAS I	PPU	ACEH	ACEH BARAT	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH
4	425874326	296852327	2004-12-30	ANAK	LAKI-LAKI	BELUM KAWIN	KELAS II	PPU	SUMATERA UTARA	TOBA SAMOSIR	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	SUMATERA UTARA

Data diatas merupakan data peserta BPJS Kesehatan tahun 2020 dengan diagnosa penyakit Tuberkulosis. Terdapat beberapa kolom/atribut yang tersedia di dalam data kepesertaan ini, diantaranya:

1. PSTV01: Nomor Peserta
2. PSTV02: Nomor Keluarga
3. PSTV03: Tanggal Lahir Peserta
4. PSTV04: Hubungan Keluarga
5. PSTV05: Jenis Kelamin
6. PSTV06: Status Perkawinan
7. PSTV07: Kelas Rawat
8. PSTV08: Segmentasi Peserta
9. PSTV09: Provinsi Tempat Tinggal Peserta
10. PSTV10: Kabupaten/Kota Tempat Tinggal Peserta
11. PSTV11: Kepemilikan Faskes
12. PSTV12: Jenis Faskes
13. PSTV13: Provinsi Fasilitas Kesehatan Peserta Terdaftar

14. PSTV14: Kabupaten/Kota Fasilitas Kesehatan Peserta Terdaftar
15. PSTV15: Bobot
16. PSTV16: Tahun Sampel
17. PSTV17: Status Kepesertaan
18. PSTV18: Tahun Meninggal

5. TB2021_kepesertaan.dta: Kepesertaan pasien TB tahun 2021

In [10]: `df = pd.read_stata(r"C:\Users\Septiana\Downloads\Final Project\Data Sampel Final 2022\Kontekstual TB\TB2021_kepesertaan.dta")`
`# Tampilkan 5 baris pertama`
`df.head()`

Out[10]:

	PSTV01	PSTV02	PSTV03	PSTV04	PSTV05	PSTV06	PSTV07	PSTV08	PSTV09	PSTV10	PSTV11	PSTV12	PSTV13
0	32853965	31945523	1958-11-27	SUAMI	LAKI-LAKI	KAWIN	KELAS I	PPU	ACEH	ACEH TENGAH	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH
1	8555967	8555967	1936-07-01	PESERTA	LAKI-LAKI	KAWIN	KELAS I	BUKAN PEKERJA	ACEH	ACEH TENGAH	POLRI	KLINIK PRATAMA	ACEH
2	8797049	249479	1950-07-21	SUAMI	LAKI-LAKI	KAWIN	KELAS I	BUKAN PEKERJA	ACEH	ACEH SELATAN	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH
3	83472658	54408870	2012-08-12	ANAK	PEREMPUAN	BELUM KAWIN	KELAS I	PPU	ACEH	ACEH BARAT	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH
4	425874326	296852327	2004-12-30	ANAK	LAKI-LAKI	BELUM KAWIN	KELAS II	PPU	SUMATERA UTARA	TOBA SAMOSIR	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	SUMATERA UTARA

Data diatas merupakan data peserta BPJS Kesehatan tahun 2021 dengan diagnosa penyakit Tuberkulosis. Terdapat beberapa kolom/atribut yang tersedia di dalam data kepesertaan ini, diantaranya:

1. PSTV01: Nomor Peserta
2. PSTV02: Nomor Keluarga
3. PSTV03: Tanggal Lahir Peserta
4. PSTV04: Hubungan Keluarga
5. PSTV05: Jenis Kelamin
6. PSTV06: Status Perkawinan
7. PSTV07: Kelas Rawat
8. PSTV08: Segmentasi Peserta
9. PSTV09: Provinsi Tempat Tinggal Peserta
10. PSTV10: Kabupaten/Kota Tempat Tinggal Peserta
11. PSTV11: Kepemilikan Faskes
12. PSTV12: Jenis Faskes
13. PSTV13: Provinsi Fasilitas Kesehatan Peserta Terdaftar
14. PSTV14: Kabupaten/Kota Fasilitas Kesehatan Peserta Terdaftar
15. PSTV15: Bobot
16. PSTV16: Tahun Sampel
17. PSTV17: Status Kepesertaan
18. PSTV18: Tahun Meninggal

6. TB20152021_fktpnonkapitasi.dta

In [11]: `pd.read_stata(r"C:\Users\Septiana\Downloads\Final Project\Data Sampel Final 2022\Kontekstual TB\TB20152021_fktpnonkapitasi.dta")`
 mpilkan 5 baris pertama
 ead()

Out[11]:

	PSTV01	PSTV02	PSTV15	PNK02	PNK03	PNK04	PNK05	PNK06	PNK07	PNK08	...	PNK10	PNK11	PNK12
0	93858078	93216423	7.659537	183920215Y000376	2015-02-26	2015-02-26	2015-02-27	SULAWESI SELATAN	BARRU	PEMERINTAH KABUPATEN/KOTA	...	RAWAT INAP	RITP	PBI APBN
1	93747649	346217457	1.800196	19500915Y000074	2015-09-21	2015-09-23	2015-09-23	KALIMANTAN SELATAN	TAPIN	PEMERINTAH KABUPATEN/KOTA	...	RAWAT INAP	RITP	PBI APBN
2	359887820	72989971	0.959155	250630919P000299	2019-09-26	2019-09-29	2019-09-29	PAPUA	JAYAPURA	TNI AD	...	KLINIK RAWAT INAP	RITP	PPU para
3	84126594	84126594	1.110887	326360919P001086	2019-09-09	2019-09-12	2019-09-12	JAWA TENGAH	REMBANG	PEMERINTAH KABUPATEN/KOTA	...	RAWAT INAP	RITP	PBI APBN para
4	87558937	62126532	32.451832	252721019P001142	2019-10-14	2019-10-17	2019-10-17	JAWA TENGAH	KEBUMEN	PEMERINTAH KABUPATEN/KOTA	...	RAWAT INAP	RITP	PBI APBN para

5 rows x 21 columns

Terdapat beberapa kolom/atribut yang tersedia di dalam data FKTP Non-Kapitasi ini, diantaranya:

1. PSTV01: Nomor Peserta
2. PSTV02: Nomor Keluarga
3. PSTV15: Bobot
4. PNK02: ID Kunjungan
5. PNK03: Tanggal Kunjungan
6. PNK04: Tanggal Tindakan
7. PNK05: Tanggal Pulang
8. PNK06: Provinsi Faskes
9. PNK07: Kode Kab/Kota Faskes
10. PNK08: Kepemilikan Faskes
11. PNK09: Jenis Faskes
12. PNK10: Tipe Faskes
13. PNK11: Tingkat Layanan
14. PNK12: Segmen Peserta
15. PNK13: Kode dan Nama diagnosis berdasarkan ICD-10 (3 digit)
16. PNK13A: Kode diagnosis berdasarkan ICD-10 (3 digit)
17. PNK14: Kode diagnosis (3-5 digit)
18. PNK15: Nama diagnosis
19. PNK16: Nama Tindakan
20. PNK17: Biaya Tagih
21. PNK18: Biaya Verifikasi

2.2 Menelaah Data

Berikut ini hasil data-data yang ditelaah:

1. TB2019_kepesertaan.dta

```
import pandas as pd

df = pd.read_stata(r"D:\semester 6\dami\data proyek\Data Sampel Final 2022\kontekstual TB\TB2019_kepesertaan.dta")

df.describe()
```

	PSTV01	PSTV02	PSTV03	PSTV15	PSTV16	PSTV18
count	9.496600e+04	9.496600e+04	94966	94966.000000	94966.0	4476.0
mean	1.830261e+08	1.903269e+08	1982-05-31 17:32:14.612387584	8.461007	2019.0	2019.0
min	1.394000e+03	3.687000e+03	1912-12-30 00:00:00	0.156804	2019.0	2019.0
25%	4.366800e+07	4.563971e+07	1964-07-20 06:00:00	1.154869	2019.0	2019.0
50%	8.745586e+07	9.109761e+07	1982-08-06 00:00:00	2.845975	2019.0	2019.0
75%	3.455679e+08	3.518411e+08	1999-10-11 00:00:00	7.727345	2019.0	2019.0
max	4.553809e+08	4.553826e+08	2019-07-19 00:00:00	365.923492	2019.0	2019.0
std	1.597607e+08	1.607588e+08	NaN	19.109177	0.0	0.0

Berdasarkan output `df.describe()` dari data kepesertaan BPJS 2019, dapat dilihat bahwa data cukup lengkap dan rapi. Kolom PSTV01 dan PSTV02 berisi ID peserta dan keluarga tanpa nilai kosong. Tanggal lahir (PSTV03) menunjukkan rentang usia sangat luas, dari 1912 hingga 2019, dengan rata-rata sekitar tahun 1982. Bobot (PSTV15) memiliki rata-rata 8,46 dengan variasi cukup besar, yang mungkin digunakan untuk analisis representatif. Seluruh data berasal dari tahun 2019 (PSTV16), sedangkan PSTV18 hanya terisi sebagian. Secara umum, data ini siap untuk dianalisis lebih lanjut seperti segmentasi atau penghitungan usia.

2. TB2020_kepesertaan.dta

```
import pandas as pd
df = pd.read_stata(r"D:\semester 6\dami\data proyek\Data Sampel Final 2022\Kontekstual TB\TB2020_kepesertaan.dta")
df.describe()
```

	PSTV01	PSTV02	PSTV03	PSTV15	PSTV16	PSTV18
count	9.496600e+04	9.496600e+04	94966	94966.000000	94966.0	7147.000000
mean	1.830281e+08	1.903269e+08	1982-05-31 17:32:14.612387584	8.461007	2020.0	2019.370365
min	1.394000e+03	3.687000e+03	1912-12-30 00:00:00	0.156804	2020.0	2019.000000
25%	4.366800e+07	4.563971e+07	1964-07-20 06:00:00	1.154869	2020.0	2019.000000
50%	8.745586e+07	9.109761e+07	1982-08-06 00:00:00	2.845975	2020.0	2019.000000
75%	3.455679e+08	3.518411e+08	1999-10-11 00:00:00	7.727345	2020.0	2020.000000
max	4.553809e+08	4.553826e+08	2019-07-19 00:00:00	365.923492	2020.0	2020.000000
std	1.597607e+08	1.607588e+08	NaN	19.109198	0.0	0.482936

Berdasarkan hasil `df.describe()`, kolom PSTV01 dan PSTV02 berisi data numerik yang sangat kecil dengan nilai rata-rata mendekati nol dan standar deviasi rendah, menunjukkan penyebaran data yang sempit. Kolom PSTV03 kemungkinan berisi data tanggal karena rentangnya dari tahun 1912 hingga 2010. Kolom PSTV05 menunjukkan nilai dengan sebaran yang cukup besar, dengan rata-rata sekitar 54,86 dan maksimum hingga 365,53. Sementara itu, kolom PSTV16 tidak memiliki variasi karena seluruh nilainya adalah 2020. Kolom PSTV18 berisi data tahunan dengan dominasi nilai tahun 2020, namun hanya memiliki 7147 entri, jauh lebih sedikit dibanding kolom lain, sehingga perlu perhatian khusus dalam analisis.

3. TB2021_kepesertaan.dta

```
import pandas as pd
df = pd.read_stata(r"D:\semester 6\dami\data proyek\Data Sampel Final 2022\Kontekstual TB\TB2021_kepesertaan.dta")
df.describe()
```

	PSTV01	PSTV02	PSTV03	PSTV15	PSTV16	PSTV18
count	9.496600e+04	9.496600e+04	94966	94966.000000	94966.0	8654.000000
mean	1.830281e+08	1.903269e+08	1982-05-31 17:32:14.612387584	8.461007	2021.0	2019.656344
min	1.394000e+03	3.687000e+03	1912-12-30 00:00:00	0.156804	2021.0	2019.000000
25%	4.366800e+07	4.563971e+07	1964-07-20 06:00:00	1.154869	2021.0	2019.000000
50%	8.745586e+07	9.109761e+07	1982-08-06 00:00:00	2.845975	2021.0	2019.000000
75%	3.455679e+08	3.518411e+08	1999-10-11 00:00:00	7.727345	2021.0	2020.000000
max	4.553809e+08	4.553826e+08	2019-07-19 00:00:00	365.923492	2021.0	2021.000000
std	1.597607e+08	1.607588e+08	NaN	19.109198	0.0	0.759391

Berdasarkan hasil `df.describe()` dari data “TB2021_kepesertaan.dta”, terlihat bahwa data ini terdiri dari sekitar 94.966 baris dan sebagian besar kolomnya terisi lengkap. Kolom PSTV01 dan PSTV02 menyimpan ID peserta dan ID keluarga tanpa nilai kosong. Tanggal lahir peserta di kolom PSTV03 menunjukkan rentang usia yang sangat luas, dari tahun 1912 hingga 2019, dengan rata-rata tahun lahir sekitar 1982. Bobot data pada PSTV15 memiliki nilai rata-rata sekitar 8,46 dan standar deviasi cukup tinggi, yang menunjukkan adanya variasi besar antar peserta. Seluruh data berasal dari tahun 2021 (PSTV16), sedangkan PSTV18 berisi data tambahan yang hanya terisi sebagian. Secara keseluruhan, data ini cukup bersih dan bisa

langsung digunakan untuk analisis seperti segmentasi peserta atau studi distribusi usia dan bobot peserta BPJS.

4. TB20152021_fkrtl.dta

```
df = pd.read_stata(r"D:\semester 6\dami\data proyek\Data Sampel Final 2022\kontekstual TB\TB20152021_fkrtl.dta")
df.describe()
```

	PSTV01	PSTV02	PSTV15	FKL03	FKL04	FKL22	FKL32	FKL34	FKL37	FKL40
count	1.583242e+06	1.583242e+06	1.583242e+06	1583242	1583242	1.583242e+06	1.583242e+06	1.522041e+06	1.522041e+06	1.522041e+06
mean	1.246185e+08	1.425551e+08	1.020879e+01	2019-02-24 13:08:43.973466624	2019-02-25 00:12:18.899043840	3.458324e+01	7.390369e+05	1.656251e+02	7.580593e+03	2.144971e+03
min	1.394000e+03	3.687000e+03	1.568037e-01	2014-09-30 00:00:00	2015-01-01 00:00:00	1.000000e+00	6.670000e+04	0.000000e+00	0.000000e+00	0.000000e+00
25%	3.280830e+07	3.529952e+07	1.258935e+00	2018-06-08 00:00:00	2018-06-09 00:00:00	1.800000e+01	1.833000e+05	0.000000e+00	0.000000e+00	0.000000e+00
50%	6.564083e+07	7.107216e+07	3.369289e+00	2019-05-05 00:00:00	2019-05-06 00:00:00	4.400000e+01	1.904000e+05	0.000000e+00	0.000000e+00	0.000000e+00
75%	9.880540e+07	2.993567e+08	9.254922e+00	2019-12-23 00:00:00	2019-12-23 00:00:00	4.400000e+01	2.683000e+05	0.000000e+00	0.000000e+00	0.000000e+00
max	4.553809e+08	4.553826e+08	3.659235e+02	2021-12-31 00:00:00	2021-12-31 00:00:00	8.400000e+01	3.969334e+08	1.543009e+07	4.887170e+07	1.033785e+08
std	1.383672e+08	1.476768e+08	2.241520e+01	NaN	NaN	1.318115e+01	2.664309e+06	4.140777e+04	2.399153e+05	2.416425e+05

```
df = pd.read_stata(r"D:\semester 6\dami\data proyek\Data Sampel Final 2022\kontekstual TB\TB20152021_fkrtl.dta")
df.describe()
```

	FKL03	FKL04	FKL22	FKL32	FKL34	FKL37	FKL40	FKL43	FKL46	FKL47	FKL48
	1583242	1583242	1.583242e+06	1.583242e+06	1.522041e+06	1.522041e+06	1.522041e+06	1.522041e+06	1.583242e+06	1.583242e+06	
	2019-02-24 13:08:43.973466624	2019-02-25 00:12:18.899043840	3.458324e+01	7.390369e+05	1.656251e+02	7.580593e+03	2.144971e+03	2.275263e+03	1.093044e+03	7.525403e+05	7.522452e+05
	2014-09-30 00:00:00	2015-01-01 00:00:00	1.000000e+00	6.670000e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	6.670000e+04	0.000000e+00
	2018-06-08 00:00:00	2018-06-09 00:00:00	1.800000e+01	1.833000e+05	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	1.833000e+05	1.833000e+05
	2019-05-05 00:00:00	2019-05-06 00:00:00	4.400000e+01	1.904000e+05	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	1.904000e+05	1.904000e+05
	2019-12-23 00:00:00	2019-12-23 00:00:00	4.400000e+01	2.683000e+05	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	2.683000e+05	2.683000e+05
	2021-12-31 00:00:00	2021-12-31 00:00:00	8.400000e+01	3.969334e+08	1.543009e+07	4.887170e+07	1.033785e+08	3.332900e+06	1.131860e+07	3.969334e+08	3.969334e+08
	NaN	NaN	1.318115e+01	2.664309e+06	4.140777e+04	2.399153e+05	2.416425e+05	6.449953e+04	7.529128e+04	2.759764e+06	2.759039e+06

Berdasarkan hasil `df.describe()` dari data “TB20152021_fkrtl.dta”, terlihat bahwa data ini sangat besar dan lengkap, dengan lebih dari 1,5 juta baris. Kolom PSTV01 dan PSTV02 berisi ID peserta dan keluarga tanpa nilai kosong. Kolom PSTV15 menunjukkan bobot data dengan rata-rata sekitar 1, dan penyebaran data cukup lebar, menandakan ada variasi dalam pembobotan sampel. Kolom tanggal seperti FKL03 dan FKL04 menunjukkan bahwa data mencakup periode yang panjang dari tahun 2015 hingga akhir 2021. Sementara itu, kolom-kolom seperti FKL22, FKL32, dan FKL34 berisi data numerik dengan nilai maksimum yang sangat besar—ini bisa berkaitan dengan jumlah kunjungan, biaya, atau unit layanan kesehatan.

Beberapa kolom seperti FKL37, FKL40, hingga FKL48 memiliki nilai-nilai ekstrem yang menandakan adanya data yang sangat besar atau kasus khusus.

5. TB20152021_fkrtldxsekunder

```
df = pd.read_stata(r"D:\semester 6\dami\data proyek\Data Sampel Final 2022\kontekstual TB\TB20152021_fkrtldxsekunder.dta")
df.describe()
```

	FKL02	FKL24	FKL24A	FKL24B
count	1562211	1562211	1562211	1562211
unique	1160115	7807	1676	1672
top	452460620V009697	A162	A16	A16 Respiratory tuberculosis, not confirmed ba...
freq	16	103500	148402	148402

Berdasarkan hasil `df.describe()` dari data “TB20152021_fkrtldxsekunder.dta”, terlihat bahwa data ini berisi lebih dari 1,5 juta catatan yang semuanya lengkap, tanpa data kosong di kolom FKL02, FKL24, FKL24A, dan FKL24B. Kolom-kolom ini berisi informasi terkait diagnosis pasien, seperti kode penyakit dan deskripsinya. Terdapat ribuan jenis diagnosis berbeda dalam data ini, yang menunjukkan variasi penyakit yang cukup besar. Salah satu diagnosis yang paling sering muncul adalah kode A16, yaitu tuberkulosis saluran napas yang belum dikonfirmasi bakteri, dengan jumlah kemunculan lebih dari 148 ribu kali. Ini menunjukkan bahwa penyakit tersebut cukup umum dalam data dan bisa menjadi fokus penting untuk dianalisis lebih lanjut, misalnya untuk melihat tren penyakit atau kebutuhan penanganan medis.

6. TB20152021_fktpnonkapitasi.dta

```
df = pd.read_stata(r"D:\semester 6\dami\data proyek\Data Sampel Final 2022\kontekstual TB\TB20152021_fktpnonkapitasi.dta")
df.describe()
```

	PSTV01	PSTV02	PSTV15	PNK03	PNK04	PNK05	PNK17	PNK18
count	3.649300e+04	3.649300e+04	36493.000000	36493	36493	36493	3.649300e+04	3.649300e+04
mean	1.033491e+08	1.259502e+08	10.189317	2019-05-12 18:54:08.908009728	2019-05-12 21:31:18.968569344	2019-05-12 23:15:29.372756224	1.353351e+05	1.353351e+05
min	6.563800e+04	6.893000e+04	0.376170	2014-12-31 00:00:00	2014-12-31 00:00:00	2015-01-01 00:00:00	0.000000e+00	0.000000e+00
25%	3.120950e+07	3.389082e+07	1.516612	2018-03-24 00:00:00	2018-03-24 00:00:00	2018-03-24 00:00:00	2.500000e+04	2.500000e+04
50%	6.113467e+07	6.568918e+07	4.347620	2019-07-13 00:00:00	2019-07-13 00:00:00	2019-07-13 00:00:00	4.500000e+04	4.500000e+04
75%	8.974764e+07	9.855116e+07	10.225730	2020-10-02 00:00:00	2020-10-02 00:00:00	2020-10-02 00:00:00	1.200000e+05	1.200000e+05
max	4.553364e+08	4.553826e+08	324.528839	2021-12-31 00:00:00	2021-12-31 00:00:00	2021-12-31 00:00:00	5.625000e+06	5.625000e+06
std	1.224665e+08	1.390374e+08	21.182901	NaN	NaN	NaN	2.245414e+05	2.245414e+05

Berdasarkan hasil `df.describe()` dari data “TB20152021_fktpnonkapitasi.dta”, total data terdiri sebanyak 36.493 baris. Kolom PSTV01 dan PSTV02 berisi ID peserta dan ID keluarga tanpa data yang hilang. Kolom PSTV15 menunjukkan bobot sampel dengan rata-rata sekitar 10, tetapi ada penyebaran data yang cukup besar, yang artinya ada perbedaan besar antar nilai bobot. Tanggal-tanggal pada kolom PNK03 sampai PNK05 menunjukkan bahwa sebagian besar data berasal dari sekitar tahun 2019, dengan rentang waktu dari 2014 sampai 2021. Sementara itu, kolom PNK17 dan PNK18 kemungkinan berkaitan dengan hal-hal seperti biaya atau data administratif lainnya, karena nilainya sangat besar dan bervariasi. Secara

keseluruhan, data ini sudah siap untuk dianalisis lebih lanjut, misalnya untuk menghitung biaya atau mengevaluasi layanan kesehatan.

2.3 Memvalidasi Data

2.3.1 Pemeriksaan Data

Berikut ini hasil pemeriksaan data yang dilakukan pada setiap dataset:

1. Data Kepesertaan

Data kepesertaan dibentuk dengan menggabungkan seluruh data kepesertaan pasien tuberkulosis tahun 2019, 2020, dan 2021. Berikut tahap penggabungan data kepesertaan (df_kepesertaan_TB2019, df_kepesertaan_TB2020, dan df_kepesertaan_TB2021).

```
In [39]: # Menggabungkan Data Kepesertaan
# Menambahkan kolom 'Tahun' untuk membedakan data kepesertaan dari masing-masing tahun
df_kepesertaan_TB2019['Tahun'] = 2019
df_kepesertaan_TB2020['Tahun'] = 2020
df_kepesertaan_TB2021['Tahun'] = 2021
```

```
In [40]: # Menggabungkan ketiga dataset kepesertaan menjadi satu dataset
df_kepesertaan = pd.concat([df_kepesertaan_TB2019, df_kepesertaan_TB2020, df_kepesertaan_TB2021], ignore_index=True)
```

```
In [41]: df_kepesertaan.head()
```

```
Out[41]:
```

	PSTV01	PSTV02	PSTV03	PSTV04	PSTV05	PSTV06	PSTV07	PSTV08	PSTV09	PSTV10	PSTV11	PSTV12	PSTV13	PSTV14
0	21611150	21611150	1957-09-12	PESERTA	PEREMPUAN	KAWIN	KELAS I	PPU	ACEH	ACEH BESAR	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	ACEH
1	94343049	96772919	1961-12-03	SUAMI	LAKI-LAKI	KAWIN	KELAS I	PPU	ACEH	ACEH TENGGARA	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	ACEH TENGGARA
2	83393824	298548714	2002-10-05	ANAK	LAKI-LAKI	BELUM KAWIN	KELAS I	PPU	ACEH	ACEH TENGGARA	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	ACEH TENGGARA
3	328537885	328537885	1989-07-13	PESERTA	PEREMPUAN	BELUM KAWIN	KELAS III	PBI APBD	ACEH	ACEH TAMIANG	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	ACEH TAMIANG
4	67805935	67805935	1972-11-13	PESERTA	LAKI-LAKI	KAWIN	KELAS I	PPU	ACEH	ACEH TAMIANG	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	ACEH TAMIANG

```
In [43]: # Memeriksa nilai yang hilang (missing values)
df_kepesertaan.isnull().sum()
```

```
Out[43]: PSTV01      0
PSTV02      0
PSTV03      0
PSTV04      0
PSTV05      0
PSTV06      0
PSTV07      0
PSTV08      0
PSTV09      0
PSTV10      0
PSTV11      0
PSTV12      0
PSTV13      0
PSTV14      0
PSTV15      0
PSTV16      0
PSTV17      0
PSTV18    264621
Tahun        0
dtype: int64
```

Untuk validasi data pemeriksaan nilai yang hilang (missing values), ditemukan di atribut PSTV18 (Tahun Meninggal) bahwa ada 264621 data yang hilang.

2. TB20152021_fkrtl.dta

```
df = pd.read_stata(r"D:\semester 6\dami\data proyek\Data Sampel Final 2022\Kontekstual TB\TB20152021_fkrtl.dta")

# Jumlah missing value tiap kolom
print("Data dengan missing value:\n", df.isnull().sum())

# Jumlah baris duplikat
print("Data yang duplikat:", df.duplicated().sum())
```

Data dengan missing value:

PSTV01	0
PSTV02	0
PSTV15	0
FKP02	0
FKL02	0
FKL03	0
FKL04	0
FKL05	0
FKL06	0
FKL07	0
FKL08	0
FKL09	0
FKL10	0
FKL11	0
FKL12	0
FKL13	0
FKL14	0
FKL15	0
FKL15A	0
FKL16	0
FKL16A	0
FKL17	0
FKL17A	0
FKL18	0
FKL18A	0
FKL19	0
FKL19A	0
FKL20	0
FKL21	0
FKL22	0
FKL23	0
FKL25	1
FKL26	1
FKL27	0
FKL28	0
FKL29	0
FKL30	0
FKL31	0
FKL32	0
FKL33	0
FKL34	61201
FKL35	0
FKL36	0
FKL37	61201
FKL38	0
FKL39	0
FKL40	61201
FKL41	0
FKL42	0
FKL43	61201
FKL44	0
FKL45	0
FKL46	61201
FKL47	0
FKL48	0

dtype: int64
Data yang duplikat: 0

Dari hasil cek data untuk data pelayanan fkrtl , terlihat bahwa di variabel/atribut FKL34, FKL37, FKL40, FKL43, FKL46 masing-masing memiliki 61201 missing values. Untuk data tersebut tidak memiliki data duplikat.

3. TB20152021_fkrtldxsekunder

```
df = pd.read_stata(r"D:\semester 6\dami\data proyek\Data Sampel Final 2022\Kontekstual TB\TB20152021_fkrtldxsekunder.dta")

# Jumlah missing value tiap kolom
print("Data dengan missing value:\n",df.isnull().sum())

# Jumlah baris duplikat
print("Data yang duplikat:", df.duplicated().sum())

Data dengan missing value:
FKL02      0
FKL24      0
FKL24A     0
FKL24B     0
dtype: int64
Data yang duplikat: 0
```

Dari hasil cek data untuk data pelayanan fkrtl sekunder, terlihat bahwa variabel/atribut tidak memiliki missing values. Untuk data tersebut tidak memiliki data duplikat.

4. TB20152021_fktpnonkapitasi.dta

```
df = pd.read_stata(r"D:\semester 6\dami\data proyek\Data Sampel Final 2022\Kontekstual TB\TB20152021_fktpnonkapitasi.dta")

# Jumlah missing value tiap kolom
print("Data dengan missing value:\n",df.isnull().sum())

# Jumlah baris duplikat
print("Data yang duplikat:", df.duplicated().sum())
df[df.duplicated()].head(487)
```

Data dengan missing value:

PSTV01 0
PSTV02 0
PSTV15 0
PNK02 0
PNK03 0
PNK04 0
PNK05 0
PNK06 0
PNK07 0
PNK08 0
PNK09 0
PNK10 0
PNK11 0
PNK12 0
PNK13 2
PNK13A 0
PNK14 0
PNK15 0
PNK16 0
PNK17 0
PNK18 0
dtype: int64
Data yang duplikat: 487

	PSTV01	PSTV02	PSTV15	PNK02	PNK03	PNK04	PNK05	PNK06	PNK07	PNK08	...	PNK10	PNK11	PNK12	I
83	54067796	54067796	10.554526	184990317Y000070	2017-03-18	2017-03-18	2017-03-21	SULAWESI SELATAN	PINRANG	PEMERINTAH KABUPATEN/KOTA	...	RAWAT INAP	RITP	PBI APBD	A01 T parat
101	54067796	54067796	10.554526	184990317Y000070	2017-03-18	2017-03-18	2017-03-21	SULAWESI SELATAN	PINRANG	PEMERINTAH KABUPATEN/KOTA	...	RAWAT INAP	RITP	PBI APBD	A01 T parat
111	88945792	67522524	1.507258	185230317Y002042	2017-03-14	2017-03-14	2017-03-20	SULAWESI SELATAN	SIDENRENG RAPPANG	PEMERINTAH KABUPATEN/KOTA	...	RAWAT INAP	RITP	PBI APBN	A01 T parat
117	88945792	67522524	1.507258	185230317Y002042	2017-03-14	2017-03-14	2017-03-20	SULAWESI SELATAN	SIDENRENG RAPPANG	PEMERINTAH KABUPATEN/KOTA	...	RAWAT INAP	RITP	PBI APBN	A01 T parat
133	88945792	67522524	1.507258	185230317Y002042	2017-03-14	2017-03-14	2017-03-20	SULAWESI SELATAN	SIDENRENG RAPPANG	PEMERINTAH KABUPATEN/KOTA	...	RAWAT INAP	RITP	PBI APBN	A01 T parat

Dari hasil cek data untuk data pelayanan nonkapitasi, terlihat bahwa variabel/atribut tidak memiliki missing values. Untuk data tersebut memiliki 487 data duplikat.

2.3.2 Validasi Format dan Tipe Data

Berikut ini dilakukan validasi format dan tipe data:

1. TB2019_kepesertaan.dta

```
In [25]: df_kepesertaan_TB2019.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 94966 entries, 0 to 94965
Data columns (total 18 columns):
#   Column      Non-Null Count  Dtype
---  -
0   PSTV01      94966 non-null  int32
1   PSTV02      94966 non-null  int32
2   PSTV03      94966 non-null  datetime64[ns]
3   PSTV04      94966 non-null  category
4   PSTV05      94966 non-null  category
5   PSTV06      94966 non-null  category
6   PSTV07      94966 non-null  category
7   PSTV08      94966 non-null  category
8   PSTV09      94966 non-null  category
9   PSTV10      94966 non-null  category
10  PSTV11      94966 non-null  category
11  PSTV12      94966 non-null  category
12  PSTV13      94966 non-null  category
13  PSTV14      94966 non-null  category
14  PSTV15      94966 non-null  float32
15  PSTV16      94966 non-null  int16
16  PSTV17      94966 non-null  category
17  PSTV18      4476 non-null   float64
dtypes: category(12), datetime64[ns](1), float32(1), float64(1), int16(1), int32(2)
memory usage: 4.8 MB
```

Berdasarkan output `df_kepesertaan_TB2019.info()`, dataset ini terdiri dari 94.966 baris dan 18 kolom dengan penggunaan memori 4,8 MB. Sebagian besar kolom memiliki data lengkap, kecuali PSTV18 yang hanya terisi sekitar 4,7% sehingga perlu penanganan *missing value*. Kolom PSTV01 dan PSTV02 bertipe numerik, PSTV03 bertipe tanggal, 12 kolom lainnya bertipe kategorikal (PSTV04–PSTV14 dan PSTV17), serta PSTV15 dan PSTV18 bertipe desimal. Dataset ini cukup rapi dan siap dianalisis lebih lanjut, terutama dengan fokus pada distribusi kategori, statistik numerik, dan penanganan data kosong.

2. TB2020_kepesertaan.dta

```
In [26]: df_kepesertaan_TB2020.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 94966 entries, 0 to 94965
Data columns (total 18 columns):
#   Column      Non-Null Count  Dtype
---  -
0   PSTV01      94966 non-null  int32
1   PSTV02      94966 non-null  int32
2   PSTV03      94966 non-null  datetime64[ns]
3   PSTV04      94966 non-null  category
4   PSTV05      94966 non-null  category
5   PSTV06      94966 non-null  category
6   PSTV07      94966 non-null  category
7   PSTV08      94966 non-null  category
8   PSTV09      94966 non-null  category
9   PSTV10      94966 non-null  category
10  PSTV11      94966 non-null  category
11  PSTV12      94966 non-null  category
12  PSTV13      94966 non-null  category
13  PSTV14      94966 non-null  category
14  PSTV15      94966 non-null  float32
15  PSTV16      94966 non-null  int16
16  PSTV17      94966 non-null  category
17  PSTV18      7147 non-null   float64
dtypes: category(12), datetime64[ns](1), float32(1), float64(1), int16(1), int32(2)
memory usage: 4.8 MB
```

Dataset TB2020_kepesertaan.dta berisi 94.966 baris dan 18 kolom dengan memori 4,8 MB. Semua kolom terisi penuh kecuali PSTV18 yang hanya memiliki 8.654 data (sekitar 9,1%), sehingga perlu penanganan *missing value*. Struktur data mirip dengan tahun 2019: terdapat kolom numerik, tanggal, dan 12 kolom kategorikal.

3. TB2021_kepesertaan.dta

```
In [27]: df_kepesertaan_TB2021.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 94966 entries, 0 to 94965
Data columns (total 18 columns):
#   Column      Non-Null Count  Dtype  
---  -
0    PSTV01      94966 non-null  int32   
1    PSTV02      94966 non-null  int32   
2    PSTV03      94966 non-null  datetime64[ns]
3    PSTV04      94966 non-null  category
4    PSTV05      94966 non-null  category
5    PSTV06      94966 non-null  category
6    PSTV07      94966 non-null  category
7    PSTV08      94966 non-null  category
8    PSTV09      94966 non-null  category
9    PSTV10      94966 non-null  category
10   PSTV11      94966 non-null  category
11   PSTV12      94966 non-null  category
12   PSTV13      94966 non-null  category
13   PSTV14      94966 non-null  category
14   PSTV15      94966 non-null  float32
15   PSTV16      94966 non-null  int16   
16   PSTV17      94966 non-null  category
17   PSTV18      8654 non-null   float64
dtypes: category(12), datetime64[ns](1), float32(1), float64(1), int16(1), int32(2)
memory usage: 4.8 MB
```

Dataset `df_kepesertaan_TB2021` memiliki 94.966 baris dan 18 kolom dengan ukuran memori sekitar 4,8 MB. Mayoritas kolom berisi data bertipe kategorikal, sementara kolom `PSTV03` berupa tanggal, dan beberapa kolom lainnya bertipe numerik seperti *int*, *float*, dan *datetime*. Sama seperti tahun 2019, 2020 hanya kolom `PSTV18` yang memiliki *missing values* (sekitar 9,1%), yang perlu dibersihkan sebelum analisis lanjutan.

4. fkrtd_TB20152021

```
In [28]: df_fkrtd_TB20152021.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1583242 entries, 0 to 1583241
Data columns (total 55 columns):
#   Column      Non-Null Count  Dtype  
---  -
0   PSTV01      1583242 non-null  int32  
1   PSTV02      1583242 non-null  int32  
2   PSTV15      1583242 non-null  float32 
3   FKP02       1583242 non-null  object  
4   FKL02       1583242 non-null  object  
5   FKL03       1583242 non-null  datetime64[ns]
6   FKL04       1583242 non-null  datetime64[ns]
7   FKL05       1583242 non-null  category
8   FKL06       1583242 non-null  category
9   FKL07       1583242 non-null  category
10  FKL08       1583242 non-null  category
11  FKL09       1583242 non-null  category
12  FKL10       1583242 non-null  category
13  FKL11       1583242 non-null  category
14  FKL12       1583242 non-null  category
15  FKL13       1583242 non-null  category
16  FKL14       1583242 non-null  category
17  FKL15       1583242 non-null  category
18  FKL15A      1583242 non-null  object  
19  FKL16       1583242 non-null  object  
20  FKL16A      1583242 non-null  object  
21  FKL17       1583242 non-null  category
22  FKL17A      1583242 non-null  object  
23  FKL18       1583242 non-null  object  
24  FKL18A      1583242 non-null  object  
25  FKL19       1583242 non-null  object  
26  FKL19A      1583242 non-null  object  
27  FKL20       1583242 non-null  category
28  FKL21       1583242 non-null  category
29  FKL22       1583242 non-null  int8    
30  FKL23       1583242 non-null  category
31  FKL25       1583241 non-null  category
32  FKL26       1583241 non-null  category
33  FKL27       1583242 non-null  category
34  FKL28       1583242 non-null  category
35  FKL29       1583242 non-null  category
36  FKL30       1583242 non-null  object  
37  FKL31       1583242 non-null  category
38  FKL32       1583242 non-null  int32  
39  FKL33       1583242 non-null  object  
40  FKL34       1522041 non-null  float64 
41  FKL35       1583242 non-null  object  
42  FKL36       1583242 non-null  object  
43  FKL37       1522041 non-null  float64 
44  FKL38       1583242 non-null  object  
45  FKL39       1583242 non-null  object  
46  FKL40       1522041 non-null  float64 
47  FKL41       1583242 non-null  object  
48  FKL42       1583242 non-null  object  
49  FKL43       1522041 non-null  float64
```

Dataset df_fkrtd_TB20152021 terdiri dari 1.583.242 entri dan 55 kolom, dengan sebagian besar data terisi lengkap (non-null). Tipe data yang terdapat dalam dataset ini cukup beragam, meliputi tipe numerik (int32, int8, float32, float64), kategori, waktu (datetime64[ns]), serta tipe objek (teks). Beberapa kolom seperti PSTV01, PSTV02, dan PSTV03 berisi data numerik, sementara kolom seperti FKL06, FKL10, dan FKL25 menggunakan tipe kategori yang sesuai untuk kebutuhan klasifikasi. Selain itu, kolom FKL03 dan FKL04 menyimpan informasi dalam format waktu yang dapat dimanfaatkan untuk analisis temporal. Meskipun sebagian besar kolom memiliki data yang lengkap, terdapat beberapa kolom seperti FKL27, FKL40, dan FKL41 yang memiliki data kosong, sehingga memerlukan penanganan khusus pada tahap pra-proses data.

5. fkrtdxsekunder_TB20152021

```
In [29]: df_fkrtdxsekunder_TB20152021.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1562211 entries, 0 to 1562210
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype  
---  -
0    FKL02      1562211 non-null  object 
1    FKL24      1562211 non-null  object 
2    FKL24A     1562211 non-null  object 
3    FKL24B     1562211 non-null  object 
dtypes: object(4)
memory usage: 59.6+ MB
```

DataFrame ini memiliki 1562211 baris dan 4 kolom, yaitu FKL02, FKL24, FKL24B, dan FKL24C. Keempat kolom ini bertipe object, yang biasanya berarti data berupa string atau campuran (bisa juga angka yang dibaca sebagai teks). Semua kolom memiliki jumlah nilai non-null yang sama, yaitu 1562211, artinya tidak ada data yang hilang di dataset ini. Ukuran memorinya sekitar 59.6 MB. Karena semua kolom bertipe object, kemungkinan data ini berisi informasi kategori atau identitas fasilitas/instansi secara tekstual, seperti kode fasilitas, jenis layanan, atau lokasi.

6. fktppnonkapitasi

```
In [30]: df_fktppnonkapitasi_TB20152021.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 36493 entries, 0 to 36492
Data columns (total 21 columns):
#   Column      Non-Null Count  Dtype  
---  -
0    PSTV01      36493 non-null  int32  
1    PSTV02      36493 non-null  int32  
2    PSTV15      36493 non-null  float32 
3    PNK02       36493 non-null  object  
4    PNK03       36493 non-null  datetime64[ns]
5    PNK04       36493 non-null  datetime64[ns]
6    PNK05       36493 non-null  datetime64[ns]
7    PNK06       36493 non-null  category
8    PNK07       36493 non-null  category
9    PNK08       36493 non-null  category
10   PNK09       36493 non-null  category
11   PNK10       36493 non-null  category
12   PNK11       36493 non-null  category
13   PNK12       36493 non-null  category
14   PNK13       36491 non-null  category
15   PNK13A      36493 non-null  object  
16   PNK14       36493 non-null  object  
17   PNK15       36493 non-null  object  
18   PNK16       36493 non-null  category
19   PNK17       36493 non-null  int32  
20   PNK18       36493 non-null  int32  
dtypes: category(9), datetime64[ns](3), float32(1), int32(4), object(4)
memory usage: 3.3+ MB
```

DataFrame ini berisi 36493 baris dan 21 kolom. Struktur data lebih kompleks dibandingkan dataset sebelumnya. Kolom seperti PSTV01, PSTV02, PSTV15, PNK17, dan PNK18 bertipe integer, menunjukkan data numerik diskrit (kemungkinan kode atau jumlah). Kolom PSTV15 bertipe float32, mengindikasikan nilai desimal (bisa jadi total biaya, skor, atau rasio). Ada juga kolom bertipe datetime64, seperti PNK03 hingga PNK06, yang merepresentasikan tanggal/waktu. Selain itu, ada banyak kolom bertipe category, yang sangat berguna untuk efisiensi memori bila data berisi kategori berulang. Data ini memakan memori sekitar 3.3 MB, jauh lebih kecil karena pemanfaatan tipe category.

2.3.3 Korelasi Heatmap

2.3.3.1 Korelasi Heatmap Data Kepesertaan

Corelation Heatmap - Data Kepesertaan

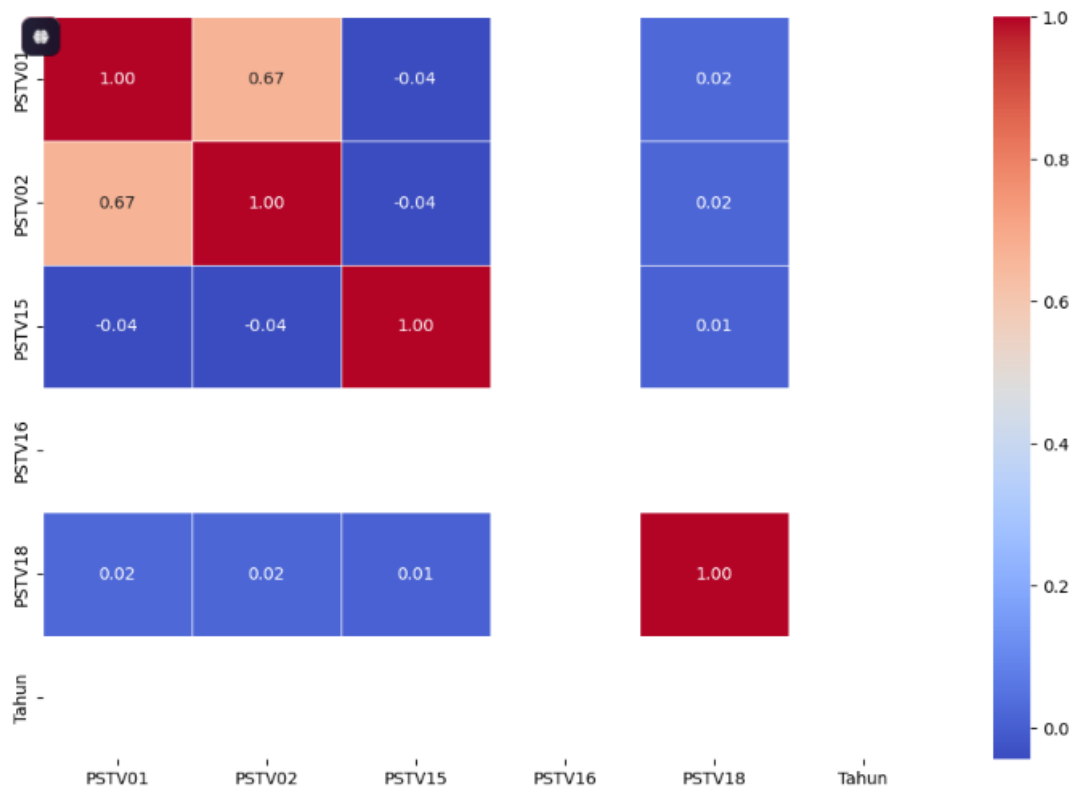
```
In [76]: import seaborn as sns  
import matplotlib.pyplot as plt
```

```
In [77]: df_kepesertaan_numeric = df_kepesertaan.select_dtypes(include=['number'])
```

```
In [78]: correlation_matrix = df_kepesertaan_numeric.corr()
```

```
In [79]: plt.figure(figsize=(12, 8))  
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
```

Out[79]: <Axes: >



Dari hasil korelasi heatmap diatas, dapat dilihat bahwa:

- Terdapat hubungan positif yang cukup kuat antara PSTV01 (Nomor Peserta) dan PSTV02 (Nomor Keluarga)
- Tidak ada hubungan yang signifikan antara PSTV01 (Nomor Peserta) dan PSTV15 (Bobot)
- Tidak ada korelasi yang berarti antara PSTV01 (Nomor Peserta) dan PSTV16 (Tahun Sampel)
- Tidak ada hubungan yang signifikan antara PSTV01 (Nomor Peserta) dan PSTV18 (Tahun Meninggal)
- Tidak ada hubungan nyata antara PSTV15 dalam hal bobot sampel dan nomor keluarga (PSTV02)

- Tidak ada korelasi yang signifikan antara PSTV02 (Nomor Keluarga) dan PSTV16 (Tahun Sampel)
- Tidak ada hubungan yang jelas antara PSTV02 (Nomor Keluarga) dan PSTV18 (Tahun Meninggal).
- Bobot sampel peserta (PSTV15) tidak memiliki hubungan yang signifikan dengan PSTV16 (Tahun Sampel).
- Bobot sampel peserta (PSTV15) tidak berkorelasi dengan PSTV18 (Tahun Meninggal).
- Tidak ada korelasi yang signifikan antara PSTV16 (Tahun Sampel) dan PSTV18 (Tahun Meninggal)
- Tahun (PSTV16) dan kolom lainnya (PSTV01, PSTV02, PSTV15, PSTV18) sangat rendah (sekitar 0.02 hingga 0.01), yang menunjukkan bahwa tidak ada hubungan yang signifikan antara tahun data dan kolom lainnya.

2.3.3.2 Korelasi Heatmap Data FKTP Nom-Kapitasi

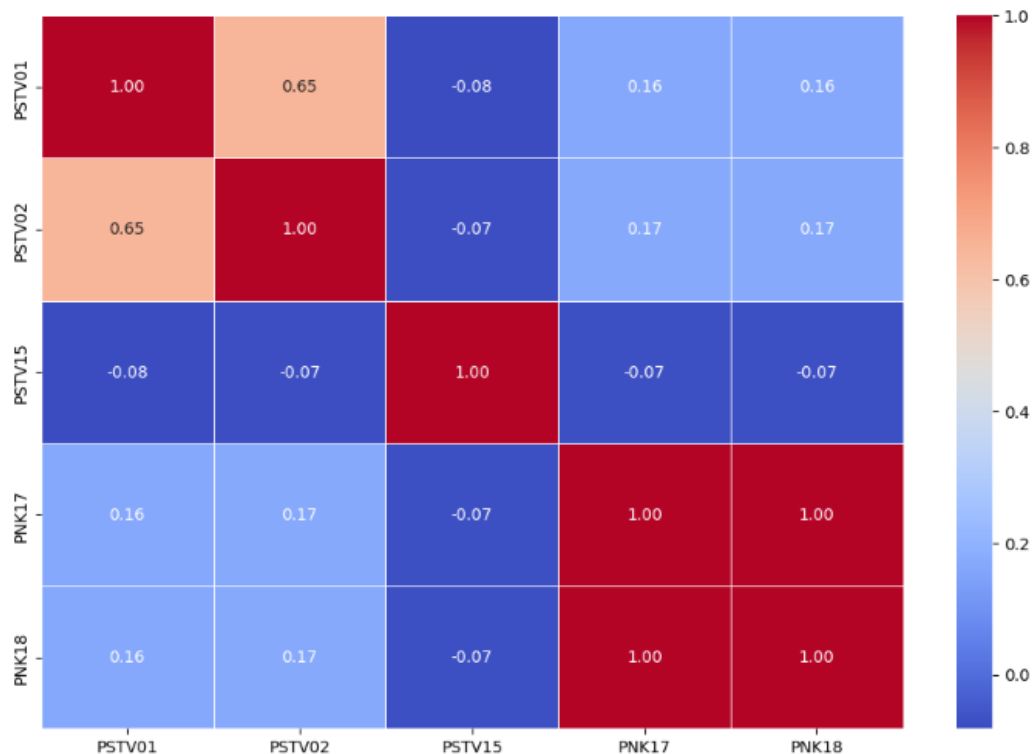
Corelation Heatmap - Data FKTP Non-Kapitasi

```
In [98]: df_fktpnonkapitasi_TB20152021_numeric = df_fktpnonkapitasi_TB20152021.select_dtypes(include=['number'])
```

```
In [99]: correlation_matrix = df_fktpnonkapitasi_TB20152021_numeric.corr()
```

```
In [100]: plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
```

```
Out[100]: <Axes: >
```



Dari hasil korelasi heatmap diatas, dapat dilihat bahwa:

- Terdapat hubungan positif yang cukup kuat antara PSTV01 (Nomor Peserta) dan PSTV02 (Nomor Keluarga)
- Terdapat hubungan negatif yang cukup kuat antara PSTV01 (Nomor Peserta) dan PSTV15
- Perubahan pada PSTV01 tidak terlalu mempengaruhi PNK17.
- Perubahan pada PSTV01 tidak terlalu mempengaruhi PNK18.
- Korelasi negatif yang sangat lemah ini menunjukkan bahwa tidak ada hubungan yang jelas antara PSTV02 dan PSTV15
- Korelasi positif yang sangat lemah ini menunjukkan sedikit hubungan antara PSTV02 dan PNK17.
- Korelasi positif yang sangat lemah ini menunjukkan sedikit hubungan antara PSTV02 dan PNK18.
- Korelasi negatif yang sangat lemah ini menunjukkan bahwa tidak ada hubungan yang jelas antara PSTV15 dan PNK17.
- Korelasi negatif yang sangat lemah ini menunjukkan bahwa antara PSTV15 dan PNK18, hampir tidak ada hubungan yang signifikan
- Korelasi positif sempurna ini menunjukkan bahwa PNK17 dan PNK18 bergerak bersama secara identik.

2.3.3.3 Korelasi Heatmap Data FKRTL

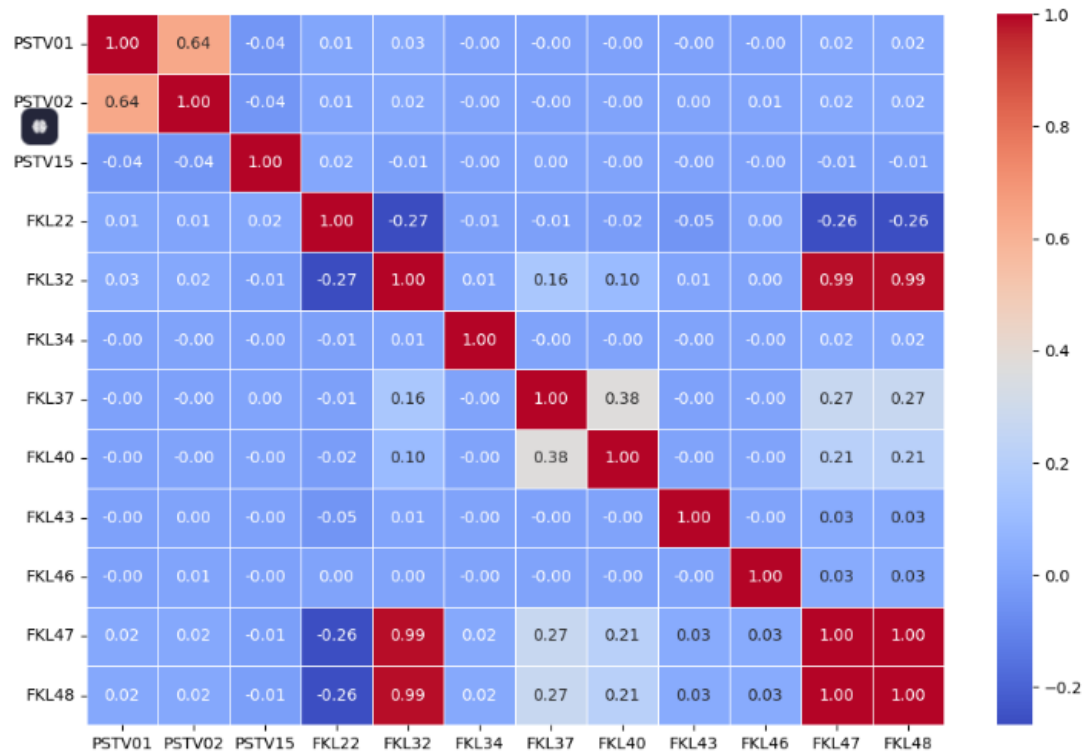
Corelation Heatmap - Data FKRTL

```
In [95]: df_fkrtl_TB20152021_numeric = df_fkrtl_TB20152021.select_dtypes(include=['number'])

In [96]: correlation_matrix = df_fkrtl_TB20152021_numeric.corr()

In [97]: plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

Out[97]: <Axes: >
```



Dari hasil korelasi heatmap diatas, dapat dilihat bahwa:

- Korelasi positif yang cukup kuat ini menunjukkan bahwa ketika nilai PSTV01 meningkat, kemungkinan besar nilai PSTV02 juga akan meningkat.
- Korelasi negatif yang sangat lemah ini menunjukkan bahwa hampir tidak ada hubungan antara PSTV01 dan PSTV15.
- Korelasi negatif yang sangat lemah ini menunjukkan bahwa tidak ada hubungan yang signifikan antara PSTV02 dan PSTV15
- Perubahan pada PSTV01 hampir tidak mempengaruhi FKL22.
- Korelasi positif yang sangat lemah ini menunjukkan hubungan yang hampir tidak ada antara PSTV01 dan FKL32.
- Korelasi mendekati nol ini menunjukkan bahwa tidak ada korelasi antara PSTV01 dan FKL34.
- Korelasi positif yang sangat lemah ini menunjukkan bahwa ada hubungan yang sangat kecil antara PSTV01 dan FKL37, meskipun hampir tidak ada pengaruh yang jelas.

- Korelasi positif yang sangat lemah ini menunjukkan bahwa sedikit hubungan positif ada antara PSTV01 dan FKL40.
- : Korelasi mendekati nol ini menunjukkan bahwa tidak ada korelasi antara PSTV 01 dan FKL43
- Korelasi negatif yang sangat lemah ini menunjukkan hampir tidak ada hubungan negatif antara PSTV01 dan FKL46.
- Korelasi yang sangat kecil ini menunjukkan tidak ada hubungan yang jelas antara PSTV01 dan FKL47
- Korelasi positif yang sangat lemah ini menunjukkan bahwa ada hubungan yang sangat kecil antara PSTV01 dan FKL48, meskipun hampir tidak ada pengaruh yang jelas.
- Korelasi positif yang sangat lemah ini menunjukkan bahwa hubungan antara PSTV02 dan FKL22 sangat kecil dan hampir tidak ada pengaruhnya
- Korelasi positif yang sangat lemah ini menunjukkan bahwa ada sedikit hubungan antara PSTV02 dan FKL32, meskipun pengaruhnya sangat kecil
- Korelasi mendekati nol ini menunjukkan bahwa tidak ada korelasi yang signifikan antara PSTV02 dan FKL34.
- FKL47 dan FKL48 yang memiliki korelasi sempurna (**1.00**), yang menunjukkan bahwa kedua variabel ini bergerak bersama dengan cara yang sangat identik.
- Korelasi tinggi lainnya juga muncul antara FKL22 dan FKL47 (**0.99**), serta antara FKL32 dan FKL47 (**0.99**). Ini menunjukkan bahwa variabel-variabel ini sangat saling berhubungan.

BAB III DATA PREPARATION

3.1 Data Selection

Pemilihan data mencakup pemilihan dataset, periode waktu, serta variabel-variabel penting yang merepresentasikan karakteristik wilayah peserta Tuberkulosis (TB) secara menyeluruh.

3.1.1 Sumber Data

Data yang digunakan bersumber dari dataset kontekstual Tuberkulosis milik BPJS Kesehatan Tahun 2022, yang mencakup data kepesertaan dan pelayanan kesehatan TB dalam rentang waktu tahun 2019-2021, diantaranya:

1. df_kepesertaan.dta: Data kepesertaan Tuberkulosis tahun 2019, 2020, 2021
2. TB20152021_fkrtl.dta: Data pelayanan Tuberkulosis di rumah sakit
3. TB20152021_fktpnonkapitasi.dta: Data pelayanan Tuberkulosis di fasilitas kesehatan tingkat pertama (fktp) non-kapitasi

3.1.2 Pemilihan Variabel

Pemilihan Variabel dari data Kepesertaan

Kode	Nama Variabel	Keterangan
PSTV03	Tanggal Lahir Peserta	Dari kolom ini, kita dapat menghitung usia peserta, yang merupakan informasi demografis penting.
PSTV05	Jenis Kelamin	Menunjukkan jenis kelamin peserta yang bisa mempengaruhi pola kesehatan di wilayah.
PSTV09	Provinsi Peserta	Informasi provinsi tempat tinggal peserta, penting untuk segmentasi wilayah.
PSTV10	Kabupaten/Kota	Kolom ini juga penting untuk mengetahui tingkat detail wilayah tempat tinggal peserta.
PSTV12	Jenis Faskes	Jenis fasilitas kesehatan yang digunakan, untuk melihat tingkat akses peserta terhadap layanan kesehatan.
PSTV13	Provinsi Faskes	Menunjukkan provinsi tempat

		fasilitas kesehatan peserta terdaftar
PSTV15	Bobot	Bobot dapat menjadi indikator penting untuk analisis representasi sampel dalam klaster.
Tahun	Tahun Data	Tahun data sampel bisa memberikan informasi tentang dinamika waktu dalam distribusi peserta.

Pemilihan Variabel dari data FKRTL

Kode	Nama Variabel	Keterangan
FKL03	Tanggal Datang Kunjungan FKRTL	Tanggal kunjungan ke fasilitas FKRTL, bisa membantu mengetahui frekuensi kunjungan peserta ke fasilitas kesehatan tingkat lanjut.
FKL05	Provinsi FKRTL	Provinsi fasilitas kesehatan tempat peserta berobat, yang menunjukkan lokasi fasilitas kesehatan.
FKL06	Kabupaten/Kota FKRTL	Kabupaten/Kota fasilitas kesehatan tempat peserta berobat.
FKL07	Kepemilikan FKRTL	Jenis kepemilikan fasilitas kesehatan, ini bisa memberikan informasi mengenai aksesibilitas dan jenis fasilitas.
FKL08	Jenis FKRTL	Jenis fasilitas rujukan tingkat lanjut yang digunakan oleh peserta, yang menggambarkan tingkat fasilitas yang diterima.
FKL12	Segmen Peserta Saat Akses	Kategori peserta saat

	Layanan FKRTL	menggunakan layanan FKRTL yang memberikan gambaran jenis peserta yang menggunakan layanan.
FKL14	Status Pulang dari FKRTL	Status peserta setelah menggunakan fasilitas kesehatan, untuk mengetahui apakah layanan sudah selesai atau masih berlanjut.

Pemilihan Variabel dari data FKTP Non-Kapitasi

Kode	Nama Variabel	Keterangan
PNK03	Tanggal Kunjungan	Tanggal kunjungan ke fasilitas kesehatan tingkat pertama
PNK06	Provinsi Faskes	Provinsi tempat fasilitas kesehatan yang digunakan oleh peserta
PNK07	Kode Kab/Kota Faskes	Kabupaten/Kota tempat fasilitas kesehatan
PNK09	Jenis Faskes	Jenis fasilitas kesehatan yang digunakan oleh peserta.
PNK10	Tipe Faskes	Tipe fasilitas kesehatan
PNK12	Segmen Peserta	Segmen peserta yang menggunakan fasilitas ini

3.2 Data Cleaning

3.2.1 Menghapus Nilai yang Hilang

Melakukan penanganan untuk mengisi missing values dengan masing-masing kolom numerik menggunakan mean dan kolom kategorikal menggunakan mode.

```
In [5]: from sklearn.impute import SimpleImputer

In [6]: imputer = SimpleImputer(strategy='most_frequent')
df_kepesertaan = pd.DataFrame(imputer.fit_transform(df_kepesertaan), columns=df_kepesertaan.columns)

In [8]: df_fkrtl_TB20152021 = pd.DataFrame(imputer.fit_transform(df_fkrtl_TB20152021), columns=df_fkrtl_TB20152021.columns)
df_fktpnonkapitasi_TB20152021 = pd.DataFrame(imputer.fit_transform(df_fktpnonkapitasi_TB20152021), columns=df_fktpnonkapitasi_TB20152021.columns)
```

```
In [9]: df_kepesertaan.isnull().sum()
```

```
Out[9]: index      0
      PSTV01    0
      PSTV02    0
      PSTV03    0
      PSTV04    0
      PSTV05    0
      PSTV06    0
      PSTV07    0
      PSTV08    0
      PSTV09    0
      PSTV10    0
      PSTV11    0
      PSTV12    0
      PSTV13    0
      PSTV14    0
      PSTV15    0
      PSTV16    0
      PSTV17    0
      PSTV18    0
      Tahun      0
      dtype: int64
```



```
In [10]: df_fkrtl_TB20152021.isnull().sum()
```

```
Out[10]: PSTV01    0
          PSTV02    0
          PSTV15    0
          FKP02     0
          FKL02     0
          FKL03     0
          FKL04     0
          FKL05     0
          FKL06     0
          FKL07     0
          FKL08     0
          FKL09     0
          FKL10     0
          FKL11     0
          FKL12     0
          FKL13     0
          FKL14     0
          FKL15     0
          FKL15A    0
          FKL16     0
          FKL16A    0
          FKL17     0
          FKL17A    0
          FKL18     0
          FKL18A    0
          FKL19     0
          FKL19A    0
          FKL20     0
          FKL21     0
          FKL22     0
          FKL23     0
          FKL25     0
          FKL26     0
          FKL27     0
          FKL28     0
          FKL29     0
          FKL30     0
          FKL31     0
          FKL32     0
          FKL33     0
          FKL34     0
          FKL35     0
          FKL36     0
          FKL37     0
          FKL38     0
          FKL39     0
          FKL40     0
          FKL41     0
          FKL42     0
          FKL43     0
          FKL44     0
          FKL45     0
          FKL46     0
          FKL47     0
          FKL48     0
```

```
In [11]: df_fktpnonkapitasi_TB20152021.isnull().sum()
```

```
Out[11]: PSTV01    0
PSTV02    0
PSTV15    0
PNK02     0
PNK03     0
PNK04     0
PNK05     0
PNK06     0
PNK07     0
PNK08     0
PNK09     0
PNK10     0
PNK11     0
PNK12     0
PNK13     0
PNK13A    0
PNK14     0
PNK15     0
PNK16     0
PNK17     0
PNK18     0
dtype: int64
```

3.2.2 Menghapus Duplikat

Duplikat dapat memengaruhi hasil analisis, jadi akan dilakukan penghapusan baris yang duplikat dari ketiga dataset.

```
In [12]: df_kepesertaan = df_kepesertaan.drop_duplicates()
df_fkrtl_TB20152021 = df_fkrtl_TB20152021.drop_duplicates()
df_fktpnonkapitasi_TB20152021 = df_fktpnonkapitasi_TB20152021.drop_duplicates()
```

```
In [13]: df_kepesertaan.duplicated().sum()
```

```
Out[13]: 0
```

```
In [14]: df_fkrtl_TB20152021.duplicated().sum()
```

```
Out[14]: 0
```

```
In [15]: df_fktpnonkapitasi_TB20152021.duplicated().sum()
```

```
Out[15]: 0
```

3.2.3 Memperbaiki Format Data

Agar data dapat digunakan untuk analisis lebih lanjut, maka diperlukan memastikan setiap kolom memiliki format data yang benar. Memastikan kolom tanggal seperti Tanggal Lahir Peserta (PSTV03), Tanggal Kunjungan FKRTL (FKL03), dan Tanggal Kunjungan FKTP (PNK03) memiliki format tanggal yang benar.

```
In [17]: df_kepesertaan.dtypes
```

```
Out[17]: index                object
PSTV01                object
PSTV02                object
PSTV03    datetime64[ns]
PSTV04                object
PSTV05                object
PSTV06                object
PSTV07                object
PSTV08                object
dtype: object
```

```
In [18]: df_fkrtl_TB20152021.dtypes
```

```
Out[18]: PSTV01                object
PSTV02                object
PSTV15                object
FKP02                object
FKL02                object
FKL03    datetime64[ns]
FKL04    datetime64[ns]
FKL05                object
FKL06                object
FKL07                object
```

```
Out[20]: PSTV01                object
PSTV02                object
PSTV15                object
PNK02                object
PNK03    datetime64[ns]
PNK04    datetime64[ns]
PNK05    datetime64[ns]
PNK06                object
PNK07                object
PNK08                object
```

3.2.4 Menangani Kolom Kategorikal

Menangani Kolom Kategorikal

```
In [19]: from sklearn.preprocessing import LabelEncoder
```

```
In [20]: # Mengonversi kolom kategorikal menjadi numerik
label_encoder = LabelEncoder()
```

```
In [21]: df_kepesertaan['PSTV05'] = label_encoder.fit_transform(df_kepesertaan['PSTV05']) # Jenis Kelamin
df_kepesertaan['PSTV09'] = label_encoder.fit_transform(df_kepesertaan['PSTV09']) # Provinsi Tempat Tinggal
df_kepesertaan['PSTV10'] = label_encoder.fit_transform(df_kepesertaan['PSTV10']) # Kabupaten/Kota Tempat Tinggal
df_kepesertaan['PSTV12'] = label_encoder.fit_transform(df_kepesertaan['PSTV12']) # Jenis Faskes
df_kepesertaan['PSTV13'] = label_encoder.fit_transform(df_kepesertaan['PSTV13']) # Provinsi Faskes
```

```
In [23]: df_fkrtl_TB20152021['FKL05'] = label_encoder.fit_transform(df_fkrtl_TB20152021['FKL05']) # Provinsi FKRTL
df_fkrtl_TB20152021['FKL06'] = label_encoder.fit_transform(df_fkrtl_TB20152021['FKL06']) # Kabupaten/Kota FKRTL
df_fkrtl_TB20152021['FKL07'] = label_encoder.fit_transform(df_fkrtl_TB20152021['FKL07']) # Kepemilikan FKRTL
df_fkrtl_TB20152021['FKL08'] = label_encoder.fit_transform(df_fkrtl_TB20152021['FKL08']) # Jenis FKRTL
```

```
In [24]: df_fkrtl_TB20152021
```

```
In [25]: df_fktpnonkapitasi_TB20152021['PNK06'] = label_encoder.fit_transform(df_fktpnonkapitasi_TB20152021['PNK06']) # Provinsi Faskes
df_fktpnonkapitasi_TB20152021['PNK07'] = label_encoder.fit_transform(df_fktpnonkapitasi_TB20152021['PNK07']) # Kabupaten/Kota Faskes
df_fktpnonkapitasi_TB20152021['PNK09'] = label_encoder.fit_transform(df_fktpnonkapitasi_TB20152021['PNK09']) # Jenis Faskes
```

3.2.5 Menyaring Kolom yang Tidak Relevan

Pada tahap ini, melakukan pemilihan kolom yang relevan dengan tugas dan tidak menggunakan kolom yang tidak relevan dengan tugas.

```
In [27]: columns_kepesertaan = ['PSTV01', 'PSTV03', 'PSTV05', 'PSTV09', 'PSTV10', 'PSTV12', 'PSTV13', 'PSTV15', 'PSTV16']
df_kepesertaan_cleaned = df_kepesertaan[columns_kepesertaan]
```

Berikut tampilan data df_kepesertaan yang telah dibersihkan:

```
In [31]: df_kepesertaan_cleaned
```

Out[31]:

	PSTV01	PSTV03	PSTV05	PSTV09	PSTV10	PSTV12	PSTV13	PSTV15	PSTV16
0	21611150	1957-09-12	1	0	2	2	0	1.157796	2019
1	94343049	1961-12-03	0	0	8	2	0	1.558821	2019
2	83393824	2002-10-05	0	0	8	2	0	1.159913	2019
3	328537885	1989-07-13	1	0	6	2	0	9.436164	2019
4	67805935	1972-11-13	0	0	6	2	0	0.899331	2019
...
94961	446945417	1984-12-31	0	21	300	2	21	3.70805	2019
94962	339605943	1989-03-17	1	8	480	2	8	35.093258	2019
94963	292512825	1979-05-31	1	32	97	0	32	2.652437	2019
94964	412089355	1970-02-17	0	33	486	0	33	14.15259	2019
94965	354440880	2000-09-23	1	9	396	2	9	1.2245	2019

94966 rows x 9 columns

Berikut tampilan data df_fktpnonkapitasi yang sudah dibersihkan:

```
In [30]: columns_fktpnonkapitasi = ['PSTV01', 'PNK03', 'PNK06', 'PNK07', 'PNK09', 'PNK10', 'PNK12']
df_fktpnonkapitasi_cleaned = df_fktpnonkapitasi_TB20152021[columns_fktpnonkapitasi]
```

```
In [32]: df_fktpnonkapitasi_cleaned
```

Out[32]:

	PSTV01	PNK03	PNK06	PNK07	PNK09	PNK10	PNK12
0	93858078	2015-02-26	27	35	5	RAWAT INAP	PBI APBN
1	93747649	2015-09-21	12	440	5	RAWAT INAP	PBI APBN
2	359887820	2019-09-26	23	112	2	KLINIK RAWAT INAP	PPU
3	84126594	2019-09-09	9	377	5	RAWAT INAP	PBI APBN
4	87558937	2019-10-14	9	130	5	RAWAT INAP	PBI APBN
...
36488	107435122	2018-08-06	27	458	5	RAWAT INAP	PBI APBN
36489	447273237	2019-03-09	10	454	5	RAWAT INAP	PBI APBN
36490	80688271	2018-01-20	24	375	5	NON RAWAT INAP	PBI APBD
36491	93887676	2017-05-03	24	375	5	NON RAWAT INAP	PBI APBD
36492	93887676	2016-08-01	24	375	5	NON RAWAT INAP	PBI APBD

36006 rows x 7 columns

Berikut tampilan data df_fkrtl yang sudah dibersihkan:

```
In [29]: columns_fkrtl = ['PSTV01', 'FKL03', 'FKL05', 'FKL06', 'FKL07', 'FKL08', 'FKL12', 'FKL14']
df_fkrtl_cleaned = df_fkrtl_TB20152021[columns_fkrtl]
```

```
In [33]: df_fkrtl_cleaned
```

Out[33]:

	PSTV01	FKL03	FKL05	FKL06	FKL07	FKL08	FKL12	FKL14
0	98834726	2015-01-21	23	121	2	2	PPU	Sehat
1	11315168	2015-01-13	23	121	2	2	PPU	Sehat
2	48014275	2016-01-02	23	121	2	2	PBIAPBN	Sehat
3	78552717	2016-01-08	23	121	2	2	PBPU	Sehat
4	48014275	2016-01-11	23	121	2	2	PBIAPBN	Sehat
...
1583237	5759906	2019-12-18	5	151	3	2	PBIAPBD	Sehat
1583238	34060166	2019-12-18	5	151	3	2	PPU	Sehat
1583239	53933488	2019-12-22	5	151	3	2	PBIAPBD	Sehat
1583240	27230440	2020-12-11	5	151	3	2	PBIAPBN	Sehat
1583241	23942287	2020-12-28	5	151	3	2	PBIAPBN	Sehat

Selanjutnya, untuk digunakan dan dianalisis lebih lanjut, ketiga data yang telah dibersihkan akan disimpan.

Menyimpan Data cLeaned

```
In [34]: df_kepesertaan_cleaned.to_csv('D:\Data Sampel Final 2022-20230608T163803Z-001\Data Sampel Final 2022\Kontekstual TB\df_kepesertaan_cleaned.csv')
df_fkrtl_cleaned.to_csv('D:\Data Sampel Final 2022-20230608T163803Z-001\Data Sampel Final 2022\Kontekstual TB\df_fkrtl_cleaned.csv')
df_fktpnonkapitasi_cleaned.to_csv('D:\Data Sampel Final 2022-20230608T163803Z-001\Data Sampel Final 2022\Kontekstual TB\df_fktpnonkapitasi_cleaned.csv')
```

3.3 Data Construct

Pada tahap ini, akan dilakukan pembuatan fitur baru yang mungkin relevan untuk analisis segmentasi wilayah BPJS Tuberkulosis. Ini melibatkan feature engineering, yaitu mengubah atau membuat fitur baru dari data yang sudah ada untuk meningkatkan performa model.

3.3.1 Menghitung Usia Peserta

Dari kolom PSTV03 (Tanggal Lahir Peserta), kita dapat menghitung usia peserta. Usia ini akan menjadi salah satu fitur penting dalam segmentasi wilayah.

Menghitung Usia Peserta:

```
In [36]: df_kepesertaan_cleaned.loc[:, 'Usia'] = (pd.to_datetime('today') - pd.to_datetime(df_kepesertaan_cleaned['PSTV03'])).dt.days //
```

3.3.2 Menambahkan Fitur Berdasarkan Waktu (Bulan dan Tahun Kunjungan) FKRTL dan FKTP Non-Kapitasi

Dari kolom Tanggal Kunjungan FKRTL (FKL03) dan Tanggal Kunjungan FKTP (PNK03), kami menambahkan fitur Bulan_Kunjungan dan Tahun_Kunjungan untuk masing-masing jenis fasilitas.

```
In [38]: # Menambahkan fitur berdasarkan bulan dan tahun dari Tanggal Kunjungan FKRTL (FKL03)
df_fkrtl_cleaned.loc[:, 'Bulan_Kunjungan_FKRTL'] = df_fkrtl_cleaned['FKL03'].dt.month
df_fkrtl_cleaned.loc[:, 'Tahun_Kunjungan_FKRTL'] = df_fkrtl_cleaned['FKL03'].dt.year
```

```
In [40]: # Menambahkan fitur berdasarkan bulan dan tahun dari Tanggal Kunjungan FKTP (PNK03)
df_fktpnonkapitasi_cleaned.loc[:, 'Bulan_Kunjungan_FKTP'] = df_fktpnonkapitasi_cleaned['PNK03'].dt.month
df_fktpnonkapitasi_cleaned.loc[:, 'Tahun_Kunjungan_FKTP'] = df_fktpnonkapitasi_cleaned['PNK03'].dt.year
```

3.3.3 Menghitung Frekuensi Kunjungan

Frekuensi kunjungan ke **FKRTL** dan **FKTP** dihitung berdasarkan jumlah kunjungan untuk setiap peserta (berdasarkan PSTV01), yang membantu memahami pola penggunaan layanan kesehatan.

```
In [41]: fkrtl_freq = df_fkrtl_cleaned.groupby('PSTV01')['FKL03'].count().reset_index(name='Frekuensi_Kunjungan_FKRTL')
df_kepesertaan_cleaned = df_kepesertaan_cleaned.merge(fkrtl_freq, on='PSTV01', how='left')

In [42]: fktp_freq = df_fktpnonkapitasi_cleaned.groupby('PSTV01')['PNK03'].count().reset_index(name='Frekuensi_Kunjungan_FKTP')
df_kepesertaan_cleaned = df_kepesertaan_cleaned.merge(fktp_freq, on='PSTV01', how='left')
```

3.3.4 Segmentasi Berdasarkan Usia

Kolom Kategori_Usia ditambahkan untuk mengelompokkan peserta menjadi tiga kategori: Muda, Dewasa, dan Lansia berdasarkan rentang usia.

Segmentasi Berdasarkan Klasifikasi Usia

```
In [43]: bins = [0, 18, 60, 100]
labels = ['Muda', 'Dewasa', 'Lansia']
df_kepesertaan_cleaned['Kategori_Usia'] = pd.cut(df_kepesertaan_cleaned['Usia'], bins=bins, labels=labels)
```

3.3.5 Menambahkan Fitur Berdasarkan Fasilitas Kesehatan

```
In [44]: df_kepesertaan_cleaned['Jenis_Faskes'] = df_kepesertaan_cleaned['PSTV12'] # Jenis Fasilitas Kesehatan yang Digunakan
df_kepesertaan_cleaned['Provinsi_Faskes'] = df_kepesertaan_cleaned['PSTV13'] # Provinsi Fasilitas Kesehatan
```

```
: df_kepesertaan_cleaned['Segmentasi_Kunjungan'] = pd.cut(df_kepesertaan_cleaned['Frekuensi_Kunjungan_FKRTL'],
    bins=[0, 1, 5, 10, 100],
    labels=['Sangat Sedikit', 'Sedikit', 'Biasa', 'Banyak'])
```

Berikut tampilan data df_kepesertaan_cleaned:

In [46]: df_kepesertaan_cleaned

Out[46]:

	PSTV01	PSTV03	PSTV05	PSTV09	PSTV10	PSTV12	PSTV13	PSTV15	PSTV16	Usia	Frekuensi_Kunjungan_FKRTL	Frekuensi_Kunjungan_FKTP	Ki
0	21611150	1957-09-12	1	0	2	2	0	1.157796	2019	67	25.0	NaN	
1	94343049	1981-12-03	0	0	8	2	0	1.556821	2019	63	17.0	NaN	
2	83393824	2002-10-05	0	0	8	2	0	1.159913	2019	22	3.0	NaN	
3	328537885	1989-07-13	1	0	6	2	0	9.436164	2019	35	7.0	1.0	
4	67805935	1972-11-13	0	0	6	2	0	0.899331	2019	52	11.0	NaN	
...
94961	448945417	1984-12-31	0	21	300	2	21	3.70805	2019	40	8.0	NaN	
94962	339805943	1989-03-17	1	8	460	2	8	35.093258	2019	36	5.0	NaN	
94963	292512825	1979-05-31	1	32	97	0	32	2.652437	2019	45	8.0	NaN	
94964	412069355	1970-02-17	0	33	486	0	33	14.15259	2019	55	1.0	NaN	
94965	354440880	2000-09-23	1	9	396	2	9	1.2245	2019	24	20.0	NaN	

14965 rows x 14 columns

Berikut tampilan data df_fkrtl_cleaned:

In [48]: df_fkrtl_cleaned

Out[48]:

	PSTV01	FKL03	FKL05	FKL06	FKL07	FKL08	FKL12	FKL14	Bulan_Kunjungan_FKRTL	Tahun_Kunjungan_FKRTL
0	98934726	2015-01-21	23	121	2	2	PPU	Sehat	1	2015
1	11315168	2015-01-13	23	121	2	2	PPU	Sehat	1	2015
2	48014275	2016-01-02	23	121	2	2	PBI APBN	Sehat	1	2016
3	78552717	2016-01-08	23	121	2	2	PBPU	Sehat	1	2016
4	48014275	2016-01-11	23	121	2	2	PBI APBN	Sehat	1	2016
...
1583237	5759906	2019-12-18	5	151	3	2	PBI APBD	Sehat	12	2019
1583238	34060166	2019-12-18	5	151	3	2	PPU	Sehat	12	2019
1583239	53933488	2019-12-22	5	151	3	2	PBI APBD	Sehat	12	2019
1583240	27230440	2020-12-11	5	151	3	2	PBI APBN	Sehat	12	2020
1583241	23942287	2020-12-28	5	151	3	2	PBI APBN	Sehat	12	2020

Berikut tampilan data df_fktpnonkapitasi_clead:

In [49]: df_fktpnonkapitasi_cleaned

Out[49]:

	PSTV01	PNK03	PNK06	PNK07	PNK09	PNK10	PNK12	Bulan_Kunjungan_FKTP	Tahun_Kunjungan_FKTP
0	93858078	2015-02-28	27	35	5	RAWAT INAP	PBI APBN	2	2015
1	93747649	2015-09-21	12	440	5	RAWAT INAP	PBI APBN	9	2015
2	359887820	2019-09-28	23	112	2	KLINIK RAWAT INAP	PPU	9	2019
3	84128594	2019-09-09	9	377	5	RAWAT INAP	PBI APBN	9	2019
4	87558937	2019-10-14	9	130	5	RAWAT INAP	PBI APBN	10	2019
...
36488	107435122	2018-08-08	27	458	5	RAWAT INAP	PBI APBN	8	2018
36489	447273237	2019-03-09	10	454	5	RAWAT INAP	PBPU	3	2019
36490	80688271	2018-01-20	24	375	5	NON RAWAT INAP	PBI APBD	1	2018
36491	93887676	2017-05-03	24	375	5	NON RAWAT INAP	PBI APBD	5	2017
36492	93887676	2018-08-01	24	375	5	NON RAWAT INAP	PBI APBD	8	2018

3.4 Labeling Data

Labelling membantu untuk mengidentifikasi kluster wilayah atau tipologi peserta berdasarkan karakteristik tertentu, seperti jenis kelamin, usia, frekuensi kunjungan, jenis fasilitas kesehatan, dan lainnya

3.4.1 Menentukan Label Berdasarkan Usia

Pada tahap ini dilakukan pembuatan label data untuk masing-masing dataset yang digunakan. Di Data Construct sebelumnya, telah kami lakukan penambahan atribut baru yang relevan dengan tugas ini, diantaranya:

- Kategori Usia: Membagi peserta ke dalam kategori Muda, Dewasa, dan Lansia berdasarkan usia mereka.
- Segmentasi Kunjungan FKRTL: Kategorisasi berdasarkan frekuensi kunjungan ke FKRTL, seperti Sangat Sedikit, Sedikit, Biasa, dan Banyak.
- Jenis Faskes: Menandai apakah peserta menggunakan FKRTL atau FKTP.
- Label Klaster Berdasarkan Kombinasi: Menggabungkan Kategori Usia dan Segmentasi Kunjungan FKRTL untuk mendapatkan label klaster yang lebih spesifik.

3.5 Data Integration

3.5.1 Menggabungkan Data Berdasarkan ID Peserta (PSTV01)

- Gabungkan Data Kepesertaan dengan FKRTL

```
In [48]: df_merged_fkrtl = pd.merge(df_kepesertaan_cleaned, df_fkrtl_cleaned, on='PSTV01', how='inner')
```

- Gabungkan Data Hasil Penggabungan dengan FKTP Non-Kapitasi

```
: df_final_merged = pd.merge(df_merged_fkrtl, df_fktpnonkapitasi_cleaned, on='PSTV01', how='inner')
```


	PSTV01	PSTV03	PSTV05	PSTV09	PSTV10	PSTV12	PSTV13	PSTV15	PSTV16	Usia	...	Bulan_Kunjungan_FKRTL	Tahun_Kunjungan_FKRTL	PNK03
0	328537885	1989-07-13	1	0	0	2	0	9.438184	2019	35	...	2	2021	2020-05-30
1	328537885	1989-07-13	1	0	0	2	0	9.438184	2019	35	...	3	2019	2020-05-30
2	328537885	1989-07-13	1	0	0	2	0	9.438184	2019	35	...	3	2019	2020-05-30
3	328537885	1989-07-13	1	0	0	2	0	9.438184	2019	35	...	3	2019	2020-05-30
4	328537885	1989-07-13	1	0	0	2	0	9.438184	2019	35	...	7	2019	2020-05-30

3.5.2 Memverifikasi Data Gabungan

Saat dilakukan pemeriksaan untuk data gabungan ini, terdapat kolom Segmentasi Kunjungan yang memiliki 482686 missing values dan sebanyak 1250986 data duplikat di kolom PSTV01.

```
In [53]: df_final_merged.isnull().sum()
```

```
Out[53]: PSTV01                0
PSTV03                0
PSTV05                0
PSTV09                0
PSTV10                0
PSTV12                0
PSTV13                0
PSTV15                0
PSTV16                0
Usia                  0
Frekuensi_Kunjungan_FKRTL 0
Frekuensi_Kunjungan_FKTP 0
Kategori_Usia          39
Jenis_Faskes           0
Provinsi_Faskes        0
Segmentasi_Kunjungan    482686
FKL03                  0
FKL05                  0
FKL06                  0
FKL07                  0
FKL08                  0
FKL12                  0
FKL14                  0
Bulan_Kunjungan_FKRTL  0
Tahun_Kunjungan_FKRTL  0
PNK03                  0
PNK06                  0
PNK07                  0
PNK09                  0
PNK10                  0
PNK12                  0
Bulan_Kunjungan_FKTP   0
Tahun_Kunjungan_FKTP   0
dtype: int64
```

```
In [54]: df_final_merged.duplicated(subset='PSTV01').sum()
```

```
Out[54]: 1250986
```

- Berikut ini untuk menghapus data duplikat.

```
In [56]: df_final_merged_cleaned = df_final_merged.drop_duplicates(subset='PSTV01')
```

```
In [57]: df_final_merged_cleaned.duplicated(subset='PSTV01').sum()
```

Out[57]: 0

- Berikut ini untuk menyimpan data gabungan yang telah dibersihkan untuk dapat digunakan dalam analisis selanjutnya.

```
In [60]: # Menyimpan dataset yang telah digabungkan
df_final_merged_cleaned.to_csv('D:\Data Sampel Final 2022-20230608T163803Z-001\Data Sampel Final 2022\Kontekstual TB\df_final_me
```

```
In [63]: df_final_merged_cleaned
```

Out[63]:

	PSTV01	PSTV03	PSTV05	PSTV09	PSTV10	PSTV12	PSTV13	PSTV15	PSTV16	Usia	...	Bulan_Kunjungan_FKRTL	Tahun_Kunjungan_FKRTL	f
0	328537885	1989-07-13	1	0	6	2	0	9.436164	2019	35	...	2	2021	
7	75453396	1959-08-25	0	9	59	2	9	10.173553	2019	65	...	8	2019	
11	98396595	1983-11-17	0	10	286	1	10	1.258935	2019	61	...	1	2019	
101	96362720	1963-06-05	0	10	470	2	10	7.791785	2019	61	...	2	2019	
409	13050306	1949-05-25	0	27	154	2	27	6.025955	2019	76	...	1	2017	
...
1258824	36617345	1962-05-10	0	31	453	2	31	3.182838	2019	63	...	1	2019	
1258886	408096111	1958-07-01	0	22	275	2	22	1.144417	2019	66	...	12	2019	
1258887	304128687	1992-08-09	1	10	428	2	10	4.157338	2019	32	...	1	2019	
1258898	382358291	1981-12-10	0	32	374	1	32	2.609354	2019	63	...	1	2020	
1258943	402406574	1959-09-09	0	21	467	2	21	5.218708	2019	65	...	2	2019	

7961 rows x 33 columns

BAB IV MODELLING

4.1 Membuat Model

4.1.1 Memilih Fitur yang Relevan

Berdasarkan tujuan awal proyek fitur yang relevan adalah Usia, Frekuensi Kunjungan FKRTL, dan Frekuensi Kunjungan FKTP, karena fitur ini berhubungan langsung dengan pola perilaku peserta dalam mengakses layanan kesehatan, serta karakteristik demografis mereka.

```
In [56]: # Memilih fitur numerik yang relevan untuk clustering
features = ['Usia', 'Frekuensi_Kunjungan_FKRTL', 'Frekuensi_Kunjungan_FKTP', ]

# Memilih subset data yang relevan
X = df_final_merged_cleaned[features]
```

4.1.2 Melakukan Normalisasi Data

Melakukan normalisasi data sangat penting agar setiap fitur memiliki skala yang setara. Tanpa normalisasi, fitur dengan skala besar (misalnya usia) akan mendominasi hasil clustering.

```
# Menormalisasi data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

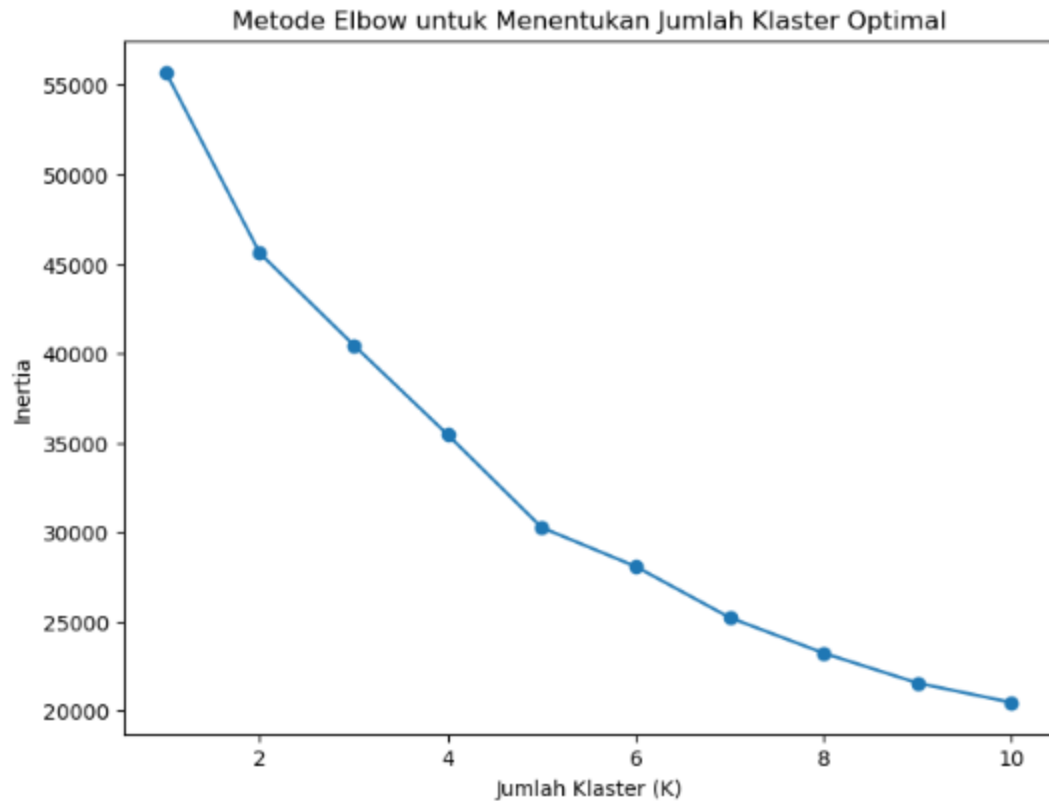
4.1.3 Menentukan Jumlah Kluster yang Optimal dengan Metode Elbow

Metode **Elbow** digunakan untuk menentukan jumlah kluster optimal dengan memplot **inertia** (jumlah kuadrat jarak dari titik data ke pusat kluster). Titik "elbow" di grafik menunjukkan jumlah kluster yang optimal.

```
]: # Menentukan jumlah kluster optimal menggunakan metode Elbow
inertia = []
k_range = range(1, 11) # Mengevaluasi jumlah kluster dari 1 sampai 10

for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled) # Melatih model K-Means
    inertia.append(kmeans.inertia_) # Menyimpan inertia untuk setiap k

# Membuat grafik Elbow untuk menentukan jumlah kluster optimal
plt.figure(figsize=(8, 6))
plt.plot(k_range, inertia, marker='o')
plt.title('Metode Elbow untuk Menentukan Jumlah Kluster Optimal')
plt.xlabel('Jumlah Kluster (K)')
plt.ylabel('Inertia')
plt.show()
```



Grafik Elbow digunakan untuk menentukan jumlah kluster yang optimal berdasarkan metode inertia. Dari grafik Elbow, terlihat bahwa titik "**elbow**" berada pada jumlah kluster **3**. Ini menunjukkan bahwa tiga kluster adalah jumlah kluster yang optimal, di mana penurunan inertia mulai melambat setelah titik tersebut.

4.1.4 Membangun Model K-Means dengan Jumlah Kluster Optimal

Setelah memilih jumlah kluster yang optimal, kita bisa membangun model **K-Means** untuk melakukan segmentasi.

```
# Berdasarkan grafik Elbow, tentukan jumlah kluster yang optimal, misalnya K = 3
optimal_k = 3

# Membangun model K-Means dengan jumlah kluster yang optimal
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
df_final_merged_cleaned['Cluster'] = kmeans.fit_predict(X_scaled) # Menambahkan Label kluster ke DataFrame

# Menampilkan hasil klasterisasi
df_final_merged_cleaned[['Usia', 'PSTV05', 'PSTV09', 'PSTV12', 'PNK09', 'Frekuensi_Kunjungan_FKRTL', 'Frekuensi_Kunjungan_FKTP', 'Cluster']].head()
```

Out[11]:

	Usia	PSTV05	PSTV09	PSTV12	PNK09	Frekuensi_Kunjungan_FKRTL	Frekuensi_Kunjungan_FKTP	Cluster
0	35	1	0	2	5	7.0	1.0	0
1	65	0	9	2	5	4.0	1.0	0
2	61	0	10	1	2	90.0	1.0	1
3	61	0	10	2	3	11.0	28.0	0
4	76	0	27	2	5	5.0	1.0	2

Dari data diatas, dapat dilihat atribut yang kami gunakan sebagai fitur dalam clustering ini diantaranya:

- Usia: Usia peserta dalam tahun.
- PSTV05: Kolom jenis kelamin
- PSTV09: Kolom provinsi tempat tinggal peserta.
- Frekuensi Kunjungan FKRTL: Menunjukkan berapa kali peserta mengunjungi fasilitas kesehatan tingkat lanjut (FKRTL)
- Frekuensi Kunjungan FKTP: menunjukkan jumlah kunjungan tambahan ke fasilitas kesehatan tingkat pertama (FKTP)
- Cluster: Kolom ini menunjukkan klaster tempat peserta tersebut dikelompokkan setelah proses klasterisasi.

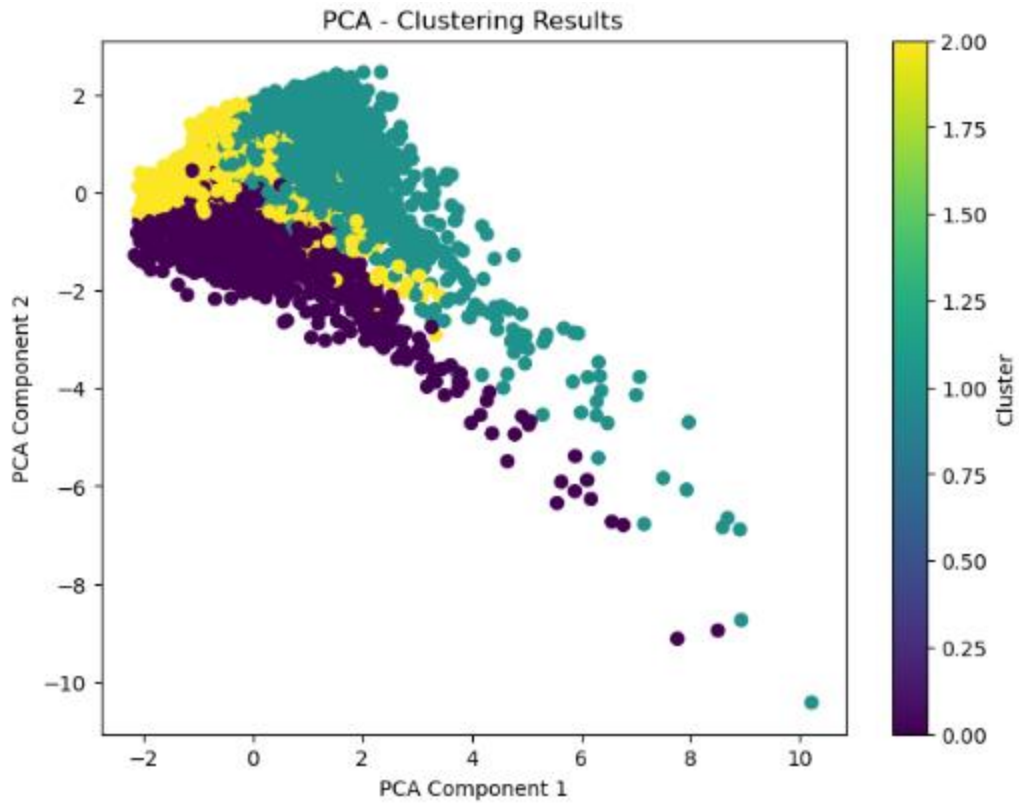
4.1.5 Visualisasi

```
] from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Reduksi dimensi menggunakan PCA ke 2 komponen utama
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# Menambahkan hasil PCA dan klaster ke DataFrame
df_final_merged_cleaned['PCA1'] = X_pca[:, 0]
df_final_merged_cleaned['PCA2'] = X_pca[:, 1]

# Visualisasi hasil clustering
plt.figure(figsize=(8, 6))
plt.scatter(df_final_merged_cleaned['PCA1'], df_final_merged_cleaned['PCA2'], c=df_final_merged_cleaned['Cluster'], cmap='viridis')
plt.title('PCA - Clustering Results')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.colorbar(label='Cluster')
plt.show()
```



Dari hasil visualisasi cluster diatas penggunaan PCA Component 1 (Sumbu X) dan PCA Component 2 (Sumbu Y) digunakan untuk mereduksi dimensi data. Data peserta BPJS Tuberkulosis telah diproyeksikan ke dalam dua dimensi ini untuk memudahkan visualisasi.

BAB V EVALUATION

5.1 Evaluasi Hasil Klasterisasi

Nilai SC 0.2745 menunjukkan bahwa klasterisasi yang terbentuk tidak memenuhi standar evaluasi yang diinginkan. Biasanya, klasterisasi dianggap memadai jika nilai SC lebih besar dari 0.55, yang menunjukkan pemisahan yang lebih jelas dan kualitas klaster yang lebih baik.

```
In [17]: silhouette_avg = silhouette_score(X_scaled, kmeans.labels_)

# Menampilkan hasil SC
print(f"Silhouette Coefficient: {silhouette_avg:.4f}")

# Jika SC > 0.55, berarti klasterisasi memenuhi syarat evaluasi
if silhouette_avg > 0.55:
    print("Clustering model memenuhi syarat evaluasi dengan SC > 55%")
else:
    print("Clustering model tidak memenuhi syarat evaluasi")

Silhouette Coefficient: 0.2745
Clustering model tidak memenuhi syarat evaluasi
```

5.2 Evaluasi Visualisasi

Setelah melakukan klasterisasi, **PCA (Principal Component Analysis)** digunakan untuk mereduksi dimensi data menjadi dua komponen utama, yang kemudian divisualisasikan dalam bentuk **scatter plot**. Warna pada plot menggambarkan klaster yang terbentuk, di mana:

- **Warna kuning (Klaster 0)**

Warna ini menunjukkan peserta yang dikelompokkan ke dalam **Klaster 0**. Peserta di klaster ini terlihat terletak di sisi kiri atas grafik, dengan komponen PCA 1 yang relatif rendah dan komponen PCA 2 yang tinggi. Ini bisa menunjukkan peserta dengan pola perilaku (usia) yang lebih sedikit mengakses fasilitas kesehatan atau memerlukan perhatian medis yang lebih sedikit.

- **Warna Torquoise (Klaster 1)**

Peserta di klaster ini menunjukkan nilai PCA Component 1 yang lebih tinggi dan PCA Component 2 yang lebih rendah. Ini menunjukkan peserta dengan usia Dewasa memiliki frekuensi kunjungan FKTP rendah dan frekuensi kunjungan FKRTL lebih tinggi (dalam beberapa kasus), yang bisa menunjukkan kebutuhan perawatan medis yang lebih intensif atau pemanfaatan layanan kesehatan yang lebih banyak.

- **Warna Ungu (Klaster 2)**

Peserta di klaster ini cenderung terletak di bagian bawah plot, dengan PCA Component 1 dan PCA Component 2 yang lebih rendah. Ini menunjukkan peserta dengan umur lebih

5.3 Relevansi Klaster dengan Kebijakan Kesehatan

Setelah melakukan evaluasi hasil klasterisasi, dapat dilakukan analisis relevansi klaster-klaster ini dengan kebijakan kesehatan yang ada. Setiap klaster memiliki karakteristik yang berbeda:

1. **Klaster 0 (Warna Kuning):**

Karakteristik: Peserta dengan sedikit akses ke fasilitas kesehatan dan risiko penyakit yang lebih rendah.

Strategi Intervensi: Fokus pada pendidikan kesehatan preventif seperti kampanye penyuluhan kesehatan, pemeriksaan rutin, dan pencegahan dini penyakit TB. Pemeriksaan tahunan atau pengecekan kesehatan rutin sangat penting untuk mendeteksi masalah sejak dini.

2. **Klaster 1 (Warna Biru Muda)**

Karakteristik: Peserta dengan usia dewasa yang lebih sering menggunakan layanan kesehatan tingkat lanjut.

Strategi Intervensi: Menyediakan perawatan lanjutan dan fokus pada manajemen penyakit untuk peserta yang membutuhkan perawatan lebih intensif. Program penyuluhan mengenai pengelolaan penyakit TB yang lebih efektif, serta peningkatan akses ke layanan kesehatan tingkat lanjut sangat dianjurkan.

3. **Klaster 2 (Warna Ungu)**

Karakteristik: Peserta yang kurang mengakses layanan kesehatan, tetapi membutuhkan perhatian medis serius ketika melakukannya.

Strategi Intervensi: Fokus pada **edukasi proaktif** untuk meningkatkan akses peserta ke layanan kesehatan lebih awal. Program **deteksi dini** penyakit TB sangat diperlukan, bersama dengan peningkatan kesadaran mengenai pentingnya memeriksakan kesehatan secara berkala.

BAB VI DEPLOYMENT

Pada tahap deployment, model klasterisasi yang telah dibangun dengan K-Means dan PCA akan diterapkan dalam bentuk aplikasi web. Aplikasi ini akan memudahkan BPJS Kesehatan untuk memprediksi klaster peserta berdasarkan data input dari formulir yang disediakan di halaman utama. Selain itu, aplikasi ini akan memberikan rekomendasi strategi intervensi yang relevan sesuai dengan klaster yang terbentuk.

6.1 Struktur dan Aplikasi Komponen Web

Aplikasi web ini menggunakan **Flask**, sebuah framework Python untuk membangun aplikasi web yang ringan dan mudah disesuaikan. Berikut adalah struktur aplikasi yang digunakan:

1. **Flask:** Framework yang digunakan untuk membuat aplikasi web yang dapat menerima input dari pengguna dan memberikan hasil prediksi berdasarkan model yang telah dilatih.
2. **HTML dan CSS:** Untuk antarmuka pengguna (UI) agar lebih mudah diakses dan digunakan.
3. **Model dan Scaler:** Model K-Means(model.pkl) dan scaler (scaler.model.pkl) digunakan untuk melakukan prediksi klaster berdasarkan data yang dimasukkan oleh pengguna.
4. **Routes:** Flask menangani dua route utama:
 - / : Menampilkan halaman formulir bagi pengguna untuk menginput data.
 - /predict: Mengambil data yang diinput, melakukan normalisasi, dan memprediksi klaster dengan model K-Means.

6.2 Langkah-Langkah Pembuatan Website

1. Aplikasi web dimulai dengan menginisialisasi aplikasi Flask dan mengimpor pustaka yang dibutuhkan, seperti joblib untuk memuat model yang telah dilatih, serta numpy dan StandarScaler untuk normalisasi data.
2. Di route /, aplikasi akan menampilkan halaman HTML yang berisi formulir untuk menginput data peserta. Halaman ini menggunakan template index.html yang terletak dalam folder templates.
3. Route /predict menangani permintaan POST dari formulir input. Data yang diinput oleh pengguna akan diproses (misalnya, normalisasi), kemudian prediksi klaster dilakukan menggunakan model yang sudah dilatih.

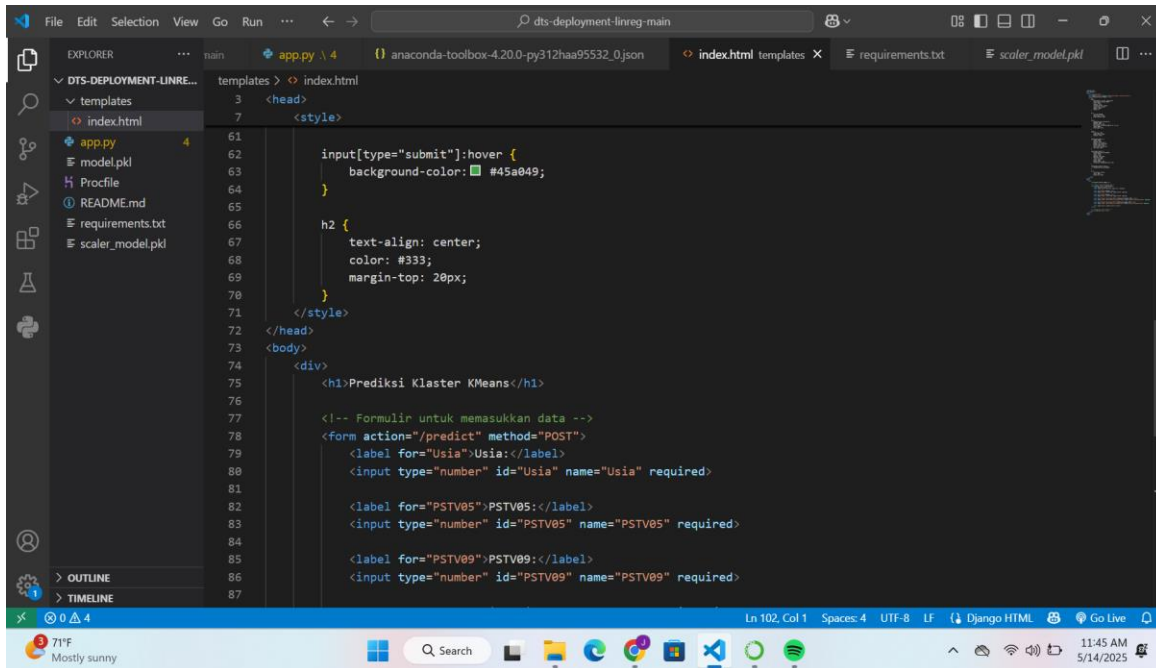
6.3 Kode Aplikasi Web

1. **Kode HTML:** Halaman utama aplikasi (index.html) yang digunakan untuk mengambil input dari pengguna. Berikut adalah cuplikan kode dari halaman index.html

```
templates > <> index.html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Prediksi Klaster KMeans</title>
7      <style>
8          body {
9              font-family: 'Arial', sans-serif;
10             background-color: #f4f4f9;
11             color: #333;
12             display: flex;
13             justify-content: center;
14             align-items: center;
15             height: 100vh;
16             margin: 0;
17         }
18
19         h1 {
20             color: #4CAF50;
21             text-align: center;
22             margin-bottom: 20px;
23         }
24
25         form {
26             background-color: #ffffff;
27             padding: 30px;
28             border-radius: 8px;
29             box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
30         }
31     </style>
32 </head>
33 <body>
34     <h1>Prediksi Klaster KMeans</h1>
35     <form>
36         <input type="text" value="Masukkan Data" />
37         <input type="button" value="Prediksi" />
38     </form>
39 </body>
40 </html>
```

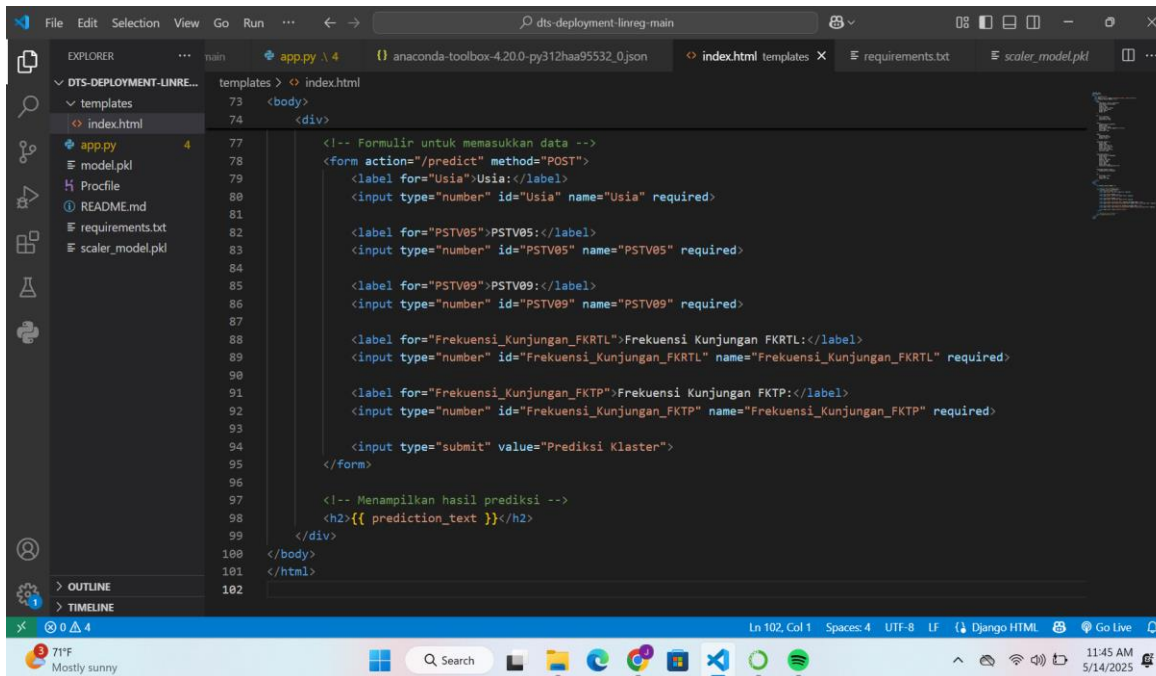
```
File Edit Selection View Go Run ... dts-deployment-linreg-main
EXPLORER templates > index.html
  templates
  index.html
  app.py
  model.pkl
  Profile
  README.md
  requirements.txt
  scaler_model.pkl

32 <style>
33
34     label {
35         font-size: 14px;
36         margin-bottom: 8px;
37         display: block;
38     }
39
40     input[type="number"] {
41         width: 100%;
42         padding: 10px;
43         margin: 8px 0 16px 0;
44         border: 1px solid #ccc;
45         border-radius: 4px;
46         box-sizing: border-box;
47         font-size: 16px;
48     }
49
50     input[type="submit"] {
51         background-color: #4CAF50;
52         color: white;
53         border: none;
54         padding: 12px 24px;
55         font-size: 16px;
56         cursor: pointer;
57         width: 100%;
58         border-radius: 4px;
59     }
60 </style>
61 </head>
62 <body>
63     <h1>Prediksi Klaster KMeans</h1>
64     <form>
65         <input type="text" value="Masukkan Data" />
66         <input type="button" value="Prediksi" />
67     </form>
68 </body>
69 </html>
```



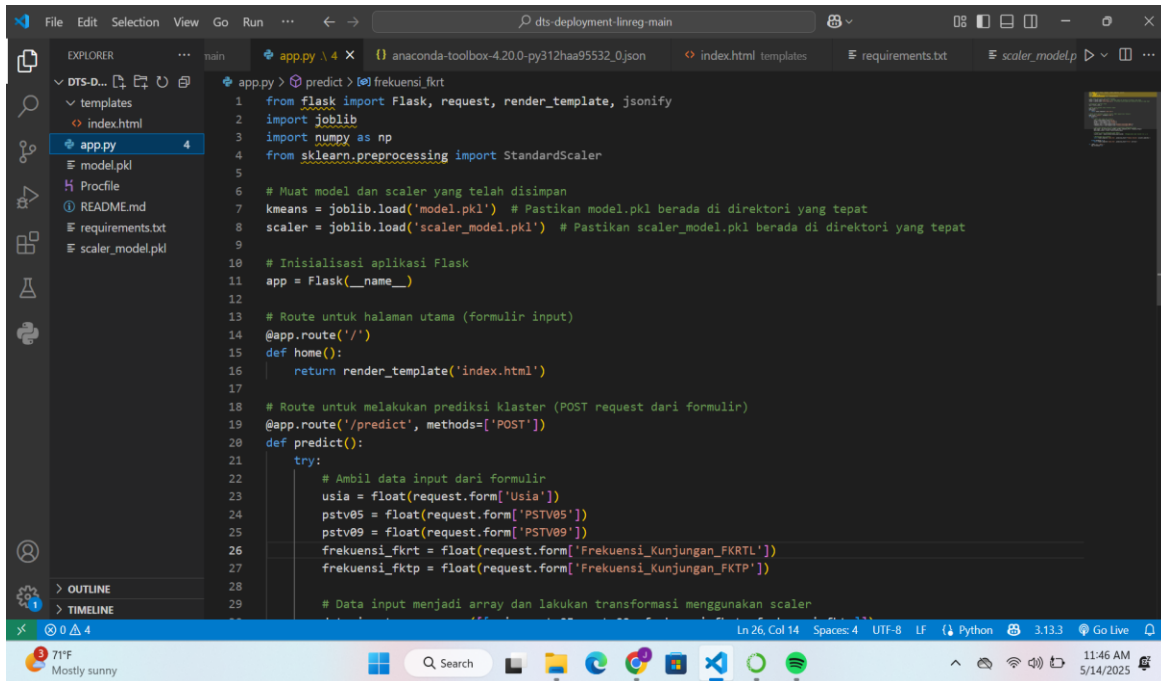
```
File Edit Selection View Go Run ... < > dts-deployment-linreg-main
EXPLORER main app.py 4 anaconda-toolbox-4.20.0-py312haa95532_0.json index.html templates x requirements.txt scaler_model.pkl
DTS-DEPLOYMENT-LINREG-main templates > index.html
  templates
    index.html
  app.py
  model.pkl
  Profile
  README.md
  requirements.txt
  scaler_model.pkl
  OUTLINE
  TIMELINE
  0 4
  71°F Mostly sunny
  Search
  11:45 AM 5/14/2025
```

```
3 <head>
4 <style>
5
6     input[type="submit"]:hover {
7         background-color: #45a049;
8     }
9
10    h2 {
11        text-align: center;
12        color: #333;
13        margin-top: 20px;
14    }
15 </style>
16 </head>
17 <body>
18 <div>
19 <h1>Prediksi Kluster KMeans</h1>
20
21 <!-- Formulir untuk memasukkan data -->
22 <form action="/predict" method="POST">
23     <label for="Usia">Usia:</label>
24     <input type="number" id="Usia" name="Usia" required>
25
26     <label for="PSTV05">PSTV05:</label>
27     <input type="number" id="PSTV05" name="PSTV05" required>
28
29     <label for="PSTV09">PSTV09:</label>
30     <input type="number" id="PSTV09" name="PSTV09" required>
31 </form>
```



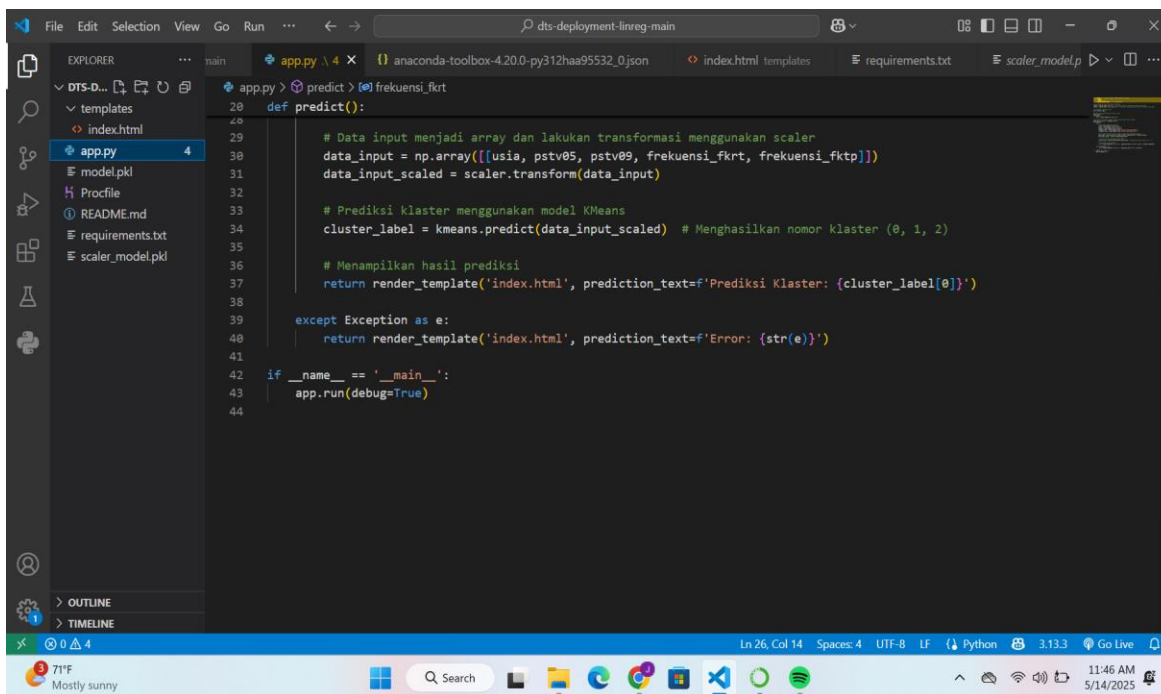
```
73 <div>
74
75 <!-- Formulir untuk memasukkan data -->
76 <form action="/predict" method="POST">
77     <label for="Usia">Usia:</label>
78     <input type="number" id="Usia" name="Usia" required>
79
80     <label for="PSTV05">PSTV05:</label>
81     <input type="number" id="PSTV05" name="PSTV05" required>
82
83     <label for="PSTV09">PSTV09:</label>
84     <input type="number" id="PSTV09" name="PSTV09" required>
85
86     <label for="Frekuensi_Kunjungan_FKRTL">Frekuensi Kunjungan FKRTL:</label>
87     <input type="number" id="Frekuensi_Kunjungan_FKRTL" name="Frekuensi_Kunjungan_FKRTL" required>
88
89     <label for="Frekuensi_Kunjungan_FKTP">Frekuensi Kunjungan FKTP:</label>
90     <input type="number" id="Frekuensi_Kunjungan_FKTP" name="Frekuensi_Kunjungan_FKTP" required>
91
92     <input type="submit" value="Prediksi Kluster">
93 </form>
94
95 <!-- Menampilkan hasil prediksi -->
96 <h2>{{ prediction_text }}</h2>
97 </div>
98 </body>
99 </html>
```

2. Kode Python (app.py): Untuk menangani request dan memuat model serta scaler yang telah dilatih.



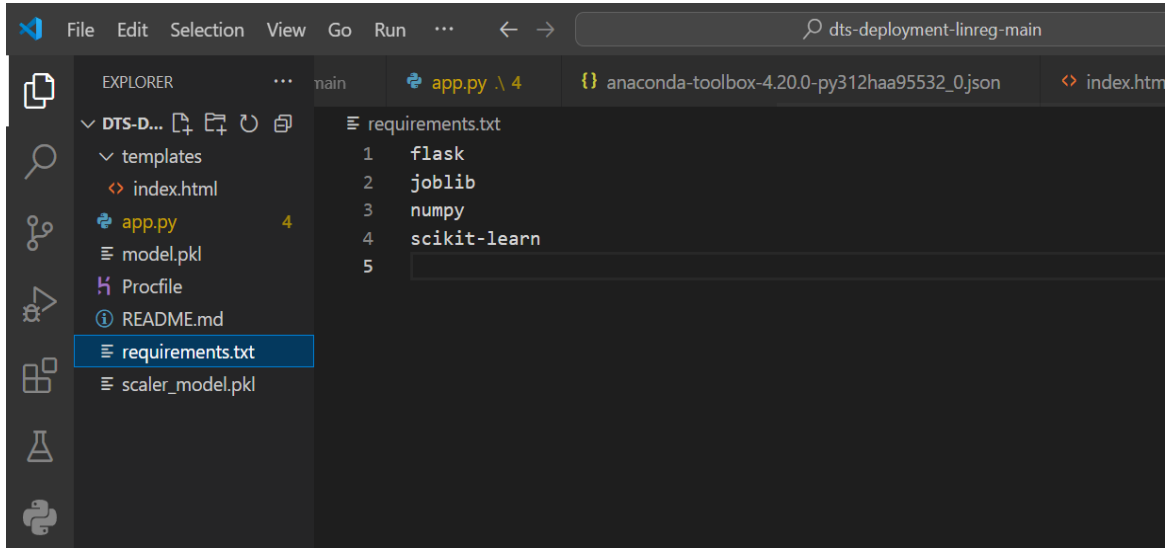
```
File Edit Selection View Go Run ... dts-deployment-linreg-main
EXPLORER main app.py \4 x {} anaconda-toolbox-4.20.0-py312haa95532_0.json index.html templates requirements.txt scaler_model.p
  DTS-D...
  templates
  index.html
  app.py 4
  model.pkl
  Profile
  README.md
  requirements.txt
  scaler_model.pkl

  app.py > predict > frekuensi_fkrt
1 from flask import Flask, request, render_template, jsonify
2 import joblib
3 import numpy as np
4 from sklearn.preprocessing import StandardScaler
5
6 # Muat model dan scaler yang telah disimpan
7 kmeans = joblib.load('model.pkl') # Pastikan model.pkl berada di direktori yang tepat
8 scaler = joblib.load('scaler_model.pkl') # Pastikan scaler_model.pkl berada di direktori yang tepat
9
10 # Inisialisasi aplikasi Flask
11 app = Flask(__name__)
12
13 # Route untuk halaman utama (formulir input)
14 @app.route('/')
15 def home():
16     return render_template('index.html')
17
18 # Route untuk melakukan prediksi kluster (POST request dari formulir)
19 @app.route('/predict', methods=['POST'])
20 def predict():
21     try:
22         # Ambil data input dari formulir
23         usia = float(request.form['Usia'])
24         pstv05 = float(request.form['PSTV05'])
25         pstv09 = float(request.form['PSTV09'])
26         frekuensi_fkrt = float(request.form['Frekuensi_Kunjungan_FKRTL'])
27         frekuensi_fktp = float(request.form['Frekuensi_Kunjungan_FKTP'])
28
29         # Data input menjadi array dan lakukan transformasi menggunakan scaler
```



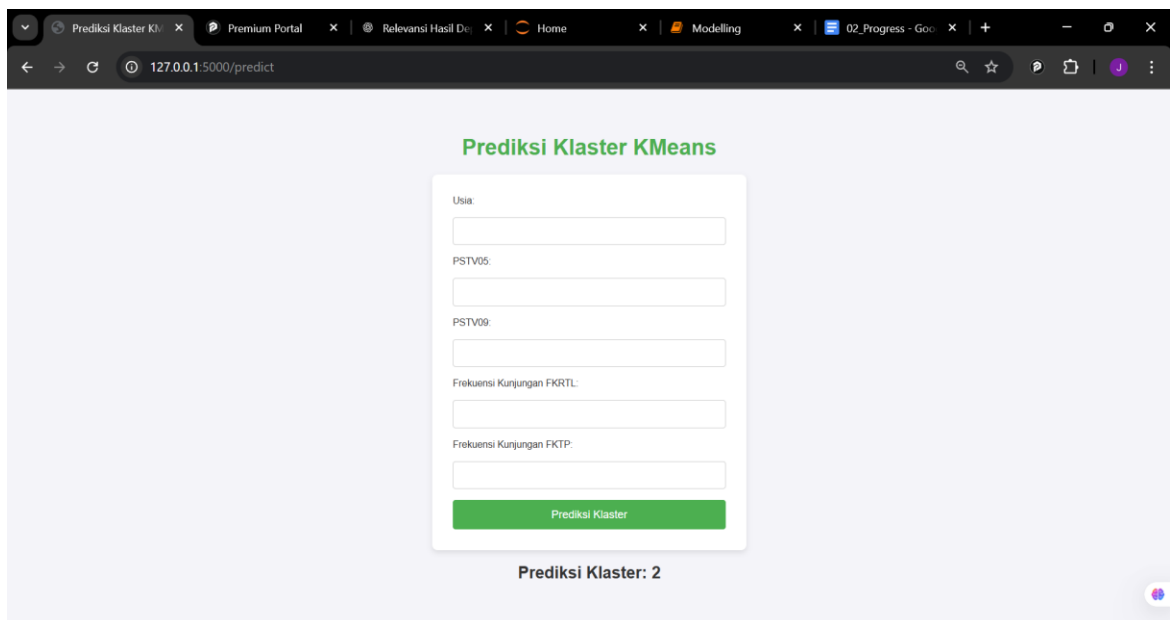
```
28 def predict():
29     # Data input menjadi array dan lakukan transformasi menggunakan scaler
30     data_input = np.array([[usia, pstv05, pstv09, frekuensi_fkrt, frekuensi_fktp]])
31     data_input_scaled = scaler.transform(data_input)
32
33     # Prediksi kluster menggunakan model KMeans
34     cluster_label = kmeans.predict(data_input_scaled) # Menghasilkan nomor kluster (0, 1, 2)
35
36     # Menampilkan hasil prediksi
37     return render_template('index.html', prediction_text=f'Prediksi Kluster: {cluster_label[0]}')
38
39 except Exception as e:
40     return render_template('index.html', prediction_text=f'Error: {str(e)}')
41
42 if __name__ == '__main__':
43     app.run(debug=True)
44
```

3. **requirements.txt**: Daftar pustaka yang dibutuhkan untuk menjalankan aplikasi web ini.



6.4 Langkah-langkah Deploy Aplikasi Web

1. Install semua pustaka yang diperlukan dengan menjalankan `pip install -r requirements.txt`
2. Setelah lingkungan siap, jalankan aplikasi Flask dengan perintah `python app.py`. Aplikasi web akan tersedia di <http://127.0.0.1:5000/>.
3. Tampilan website akan terlihat seperti pada gambar berikut



6.5 Evaluasi Website

Setelah melakukan deployment, uji aplikasi dengan berbagai input untuk memastikan hasil yang valid dan tepat.