

LAPORAN PROYEK DATA MINING

Classification of Tuberculosis Disease from 2019-2021 using Artificial Neural Network (ANN)



Disusun Oleh:

| | |
|-----------------|-------------------------------------|
| 12S20006 | Nadya Dioranta Tambunan |
| 12S20022 | Agnes Veronika Sihombing |
| 12S20044 | Hanna Dhea Christy Sihombing |

**PROGRAM STUDI SARJANA SISTEM INFORMASI
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
INSTITUT TEKNOLOGI DEL**

2023/2024

DAFTAR ISI

| | |
|--|----|
| DAFTAR ISI | 2 |
| DAFTAR GAMBAR..... | 4 |
| DAFTAR TABEL | 5 |
| BUSINESS UNDERSTANDING | 6 |
| 1.1 Objektif Bisnis..... | 6 |
| 1.2 Tujuan Bisnis..... | 7 |
| 1.3 Rencana Proyek | 7 |
| DATA UNDERSTANDING..... | 9 |
| 2.1 Mengumpulkan Data | 9 |
| 2.2 Memahami Data | 9 |
| 2.2.1 Fungsi head() | 21 |
| 2.2.2 Fungsi shape()..... | 22 |
| 2.2.3 Fungsi tail()..... | 22 |
| 2.2.4 Fungsi columns() | 23 |
| 2.2.5 Fungsi info() | 24 |
| 2.2.6 Fungsi dtypes() | 27 |
| 2.2.7 Fungsi describe() | 28 |
| 2.3 Memvalidasi Data..... | 30 |
| 2.3.1 Missing Value..... | 30 |
| 2.3.2 Duplicate Values..... | 32 |
| DATA PREPARATION..... | 33 |
| 3.1 Mengintegrasikan Data..... | 33 |
| 3.2 Data Selection..... | 34 |
| 3.3 Data Cleaning | 35 |
| 3.3.1 Feature Sellation | 35 |
| 3.3.2 Missing Value..... | 35 |
| 3.3.3 Data Duplicated | 36 |
| 3.3.4 Outlier..... | 36 |
| 3.3.5 Menampilkan proporsi kelas label..... | 37 |
| 3.3.6 Feature Correlation | 38 |
| 3.4 Menentukan Label | 39 |
| 3.5 Mengkonstruksi Data..... | 39 |
| 3.5.1 Transformasi | 39 |
| MODELLING | 41 |
| 4.1 Membangun Skenario Pengujian..... | 41 |
| 4.2 Artificial Neural Network..... | 41 |

| | |
|--|----|
| 4.3 Build Model..... | 42 |
| EVALUATION | 44 |
| DEPLOYMENT | 45 |
| 6.1 Mempersiapkan interface dan Script Code..... | 45 |
| 6.1.1 Membuat interface HTML..... | 45 |
| 6.2 Mempersiapkan Script dan Code..... | 46 |
| 6.2.1 Install Flask..... | 46 |
| 6.2.2 Create App.py | 47 |
| KESIMPULAN DAN SARAN | 50 |
| 7.1 Kesimpulan..... | 50 |
| 7.2 Saran | 50 |

DAFTAR GAMBAR

| | |
|------------------------------------|----|
| Gambar 1 Feature Correlation | 38 |
|------------------------------------|----|

DAFTAR TABEL

| | |
|--|----|
| Tabel 1 Jadwal Pelaksanaan Proyek..... | 7 |
| Tabel 2 Data fktpnonkapitasi.dta..... | 9 |
| Tabel 3 Data Kepesertaan 2019..... | 12 |
| Tabel 4 Data Kepesertaan 2020..... | 16 |
| Tabel 5 Data Kepesertaan 2021 | 18 |

BAB I

BUSINESS UNDERSTANDING

Bab ini memperjelas pemahaman konten dan kebutuhan operasi data mining dari sudut pandang bisnis. Langkah-langkahnya mencakup penetapan tujuan bisnis, pemahaman mendalam terhadap situasi bisnis, serta interpretasi tujuan bisnis atau sasaran bisnis menjadi tujuan spesifik dalam konteks operasi data mining.

1.1 Objektif Bisnis

Badan Penyelenggara Jaminan Sosial atau BPJS Kesehatan adalah badan hukum publik yang memiliki tanggung jawab terhadap penyediaan jaminan serta layanan kesehatan bagi masyarakat atau pesertanya. BPJS Kesehatan telah berhasil mengelola lebih dari 300 miliar baris data yang tersimpan dalam Big Data BPJS Kesehatan. Data ini mencakup informasi kepesertaan dan pelayanan kesehatan yang diperoleh dari masyarakat yang menggunakan Jaminan Kesehatan Nasional (JKN) untuk mengakses fasilitas pelayanan kesehatan. Sebagai upaya untuk mendorong partisipasi dan inisiatif masyarakat dalam menganalisis data tersebut, BPJS Kesehatan kemudian meluncurkan data sampel BPJS Kesehatan. Pada data sampel yang terbaru, terdapat penambahan data sampel kontekstual penyakit Tuberkulosis yang masih merupakan masalah yang signifikan di Indonesia pada saat ini. Data sampel ini merupakan pengembangan dari data sampel kontekstual pada tahun 2021 yang mengeluarkan data sampel penyakit Diabetes Mellitus.

Pada proyek ini, kami akan melakukan analisis dan klasifikasi pada data fasilitas pelayanan kesehatan peserta BPJS yang mengalami penyakit Tuberkulosis (TB) yang dapat disesuaikan berdasarkan isi PNK11. Objektif utama dalam pengerjaan proyek ini adalah melakukan prediksi untuk pasien penyakit Tuberkulosis (TB). Terdapat beberapa golongan fasilitas pelayanan kesehatan pada peserta BPJS, diantaranya:

a. RITP (Rawat Inap Tingkat Pertama)

Layanan ini mencakup rawat inap di rumah sakit tingkat pertama. RITP melibatkan pelayanan medis yang diberikan kepada peserta BPJS Kesehatan yang memerlukan perawatan inap di rumah sakit tingkat pertama.

Contoh Pelayanan: Perawatan rawat inap di rumah sakit tingkat pertama, pemeriksaan, dan pengujian medis selama masa perawatan.

b. RJTP (Rawat Jalan Tingkat Pertama)

RJTP mencakup pelayanan rawat jalan di fasilitas kesehatan tingkat pertama. Ini mencakup berbagai layanan kesehatan yang tidak memerlukan perawatan inap di rumah sakit.

Contoh Pelayanan: Kunjungan ke dokter, pemeriksaan laboratorium, pengobatan rawat jalan, konsultasi dokter, dan pelayanan kesehatan primer lainnya.

c. **PROMOTIF** (Pelayanan Kesehatan Masyarakat dan Promotif):

Layanan ini menekankan pada upaya pencegahan penyakit dan kesehatan masyarakat. **PROMOTIF** mencakup berbagai kegiatan yang bertujuan untuk meningkatkan kesehatan dan mencegah penyakit.

Contoh Pelayanan: Program vaksinasi, kampanye kesehatan, program pencegahan penyakit menular, penyuluhan kesehatan, dan kegiatan-kegiatan promosi kesehatan.

1.2 Tujuan Bisnis

Tujuan dari proyek ini yaitu ingin membangun sebuah model yang efisien terkait pengelompokkan pelayanan kesehatan BPJS untuk menganalisis dan memahami fasilitas pelayanan kesehatan pada penyakit Tuberkulosis.

1.3 Rencana Proyek

Tabel dibawah memperlihatkan tahapan perencanaan yang dilakukan dalam mencapai tujuan *data mining* dan tujuan bisnis dalam “*Classification of Tuberculosis Disease from 2019-2021 with Artificial Neural Network (ANN)*”.

Tabel 1 Jadwal Pelaksanaan Proyek

| Tahapan | Kegiatan | Waktu |
|-------------------------------|---|--------|
| Pemilihan Case/Kasus | Menentukan kasus dan algoritma. | 5 hari |
| <i>Business Understanding</i> | Menentukan objektif bisnis, tujuan bisnis, dan membuat project plan. | 3 hari |
| <i>Data Understanding</i> | Data collection, mendeskripsikan data, melakukan eksplorasi data dan memvalidasi data. | 4 hari |
| <i>Data Preparation</i> | Memilih dat, membersihkan data, membangun data, menggabungkan data dan mengintegrasikan data. | 5 hari |
| <i>Modelling</i> | Memilih teknik pemodelan, membuat Test Design, membangun model, dan menilai model | 5 hari |

| | | |
|-------------------|--|--------|
| <i>Evaluation</i> | Mengevaluasi hasil dan melakukan review proses pemodelan | 4 hari |
| <i>Deployment</i> | Membuat plan deployment, Monitoring and Maintenance Plan dan membuat laporan akhir proyek. | 7 hari |

Untuk mendukung pengerjaan setiap tahapan proses pada proyek ini, berikut adalah spesifikasi development environment pada proyek ini:

| | |
|--------------------|---|
| Tools | <ul style="list-style-type: none"> - Jupyter Notebook - Google Collab - Visual Studio Code |
| Bahasa Pemrograman | Python 3 |
| Algoritma | Artificial Neural Network (ANN) |
| Web Interface | <i>Flask Python</i> |
| Deployment Cloud | - |

BAB II DATA UNDERSTANDING

Pada bab ini akan dijelaskan mengenai pengumpulan data, mengolah data dan memvalidasi data.

2.1 Mengumpulkan Data

Langkah pertama yang dilakukan pada *data understanding* adalah pengumpulan data dalam persiapan awal dalam melakukan proyek ini yang akan digunakan pada kasus *Classification of Membership Status for BPJS Participants 2019-2021*. Dataset yang digunakan pada proyek ini terdapat pada folder Kontekstual TB, yaitu:

- 1) TB20152021_fktpnonkapitasi.dta
- 2) TB2019_kepesertaan.dta
- 3) TB2020_kepesertaan.dta
- 4) TB2021_kepesertaan.dta

2.2 Memahami Data

1. TB20152021_fktpnonkapitasi.dta

Tabel 2 Data fktpnonkapitasi.dta

| No | Atribut | Tipe Atribut | Nilai | Keterangan |
|----|---------------|--------------|----------------|---|
| 1. | <i>PSTV01</i> | int32 | Nomor peserta | Nomor identifikasi peserta yang bersifat unik dan telah diidentifikasi untuk melindungi identitas peserta sebenarnya |
| 2. | <i>PSTV02</i> | int32 | Nomor keluarga | Nomor yang mengidentifikasi kepala keluarga dalam sampel dan berfungsi sebagai penanda keluarga (peserta BPJS Kesehatan dalam satu keluarga memiliki nomor kepala keluarga yang sama) |

| | | | | |
|-----|---------------|------------------|----------------------|--|
| 3. | <i>PSTV15</i> | float32 | Bobot | Faktor pengali yang menggambarkan jumlah individu di dalam populasi diwakili oleh individu di dalam sampel |
| 4. | <i>PNK02</i> | object | ID Kunjungan | Nomor identifikasi unik untuk menandakan setiap kunjungan FKTP oleh peserta |
| 5. | <i>PNK03</i> | datetime 64[] | Tanggal kunjungan | Tanggal melakukan kunjungan |
| 6. | <i>PNK04</i> | datetime 64[] | Tanggal tindakan | Tanggal melakukan tindakan |
| 7. | <i>PNK05</i> | datetime 64[] | Tanggal pulang | Tanggal menyelesaikan kunjungan |
| 8. | <i>PNK06</i> | category | Provinsi faskes | Provinsi tempat peserta mengakses fasilitas kesehatan tingkat pratama (FKTP) |
| 9. | <i>PNK07</i> | category | Kode Kab/Kota faskes | Kabupaten/kota tempat peserta mengakses fasilitas kesehatan tingkat pratama (FKTP) |
| 10. | <i>PNK08</i> | category | Kepemilikan faskes | Kepemilikan dari fasilitas kesehatan tingkat pratama (FKTP) tempat peserta berkunjung |
| 11. | <i>PNK09</i> | category | Jenis faskes | Jenis dari fasilitas kesehatan tingkat pertama (FKTP) tempat peserta |

| | | | | |
|-----|---------------|----------|--|---|
| | | | | berkunjung |
| 12. | <i>PNK10</i> | category | Tipe faskes | Tipe dari fasilitas kesehatan tingkat pertama (FKTP) tempat peserta berkunjung |
| 13. | <i>PNK11</i> | category | Tingkat layanan | Tingkat layanan yang diterima peserta di FKTP |
| 14. | <i>PNK12</i> | category | Segmen peserta | Segmen peserta saat mengakses FKTP |
| 15. | <i>PNK13</i> | category | Kode dan Nama diagnosis berdasarkan ICD-10 (3 digit) | Kode dan nama diagnosis berdasarkan 3 digit pertama kode ICD 10 yang diperoleh dari hasil input sistem informasi BPJS Kesehatan |
| 16. | <i>PNK13A</i> | object | Kode diagnosis berdasarkan ICD-10 (3 digit) | Kode dan nama diagnosis berdasarkan 3 digit pertama kode ICD 10 yang diperoleh dari hasil input sistem informasi BPJS Kesehatan |
| 17. | <i>PNK14</i> | object | Kode diagnosis (3-5 digit) | Kode diagnosis menurut ICD 10 (jumlah digit tidak sama pada semua observasi dengan rentang 3-5 digit kode ICD 10) |
| 18. | <i>PNK15</i> | object | Nama Diagnosis | Nama diagnosis yang terbaca oleh sistem informasi BPJS Kesehatan berdasarkan kode diagnosis yang ter-input |

| | | | | |
|-----|--------------|----------|------------------|--|
| | | | | dalam sistem |
| 19. | <i>PNK16</i> | category | Nama Tindakan | Nama jenis tindakan yang dilakukan kepada pasien |
| 20. | <i>PNK17</i> | int32 | Biaya tagih | Biaya yang ditagihkan fasilitas kesehatan untuk setiap ID kunjungan |
| 21. | <i>PNK18</i> | int32 | Biaya verifikasi | Biaya yang diverifikasi BPJS Kesehatan untuk setiap nomor ID Kunjungan |

2. TB2019_kepesertaan.dta

Tabel 3 Data Kepesertaan 2019

| No | Atribut | Tipe Atribut | Label Atribut | Deskripsi |
|----|---------------|--------------|----------------|--|
| 1. | <i>PSTV01</i> | int32 | Nomor peserta | Nomor identifikasi peserta yang bersifat unik dan telah diidentifikasi untuk melindungi identitas peserta sebenarnya |
| 2. | <i>PSTV02</i> | int32 | Nomor keluarga | Nomor yang mengidentifikasi kepala keluarga dalam sampel dan berfungsi sebagai penanda keluarga (peserta BPJS Kesehatan dalam satu |

| | | | | |
|----|---------------|--------------|-----------------------|---|
| | | | | keluarga memiliki nomor kepala keluarga yang sama) |
| 3. | <i>PSTV03</i> | datetime64[] | Tanggal lahir peserta | Tanggal lahir peserta yang tercatat pada sistem BPJS Kesehatan ketika melakukan pendaftaran |
| 4. | <i>PSTV04</i> | category | Hubungan Keluarga | Hubungan atau status peserta di dalam keluarga |
| 5. | <i>PSTV05</i> | category | Jenis Kelamin | Jenis kelamin peserta |
| 6. | <i>PSTV06</i> | category | Status perkawinan | Status perkawinan |
| 7. | <i>PSTV07</i> | category | Kelas rawat | Kelas rawat peserta yang tercatat dalam sistem BPJS Kesehatan per tanggal 31 Desember 2020 |
| 8. | <i>PSTV08</i> | category | Segmentasi peserta | Segmentasi peserta yang tercatat pada sistem BPJS Kesehatan per tanggal 31 Desember 2020 |
| 9. | <i>PSTV09</i> | category | Provinsi Tempat | Provinsi tempat |

| | | | | |
|-----|---------------|----------|--|--|
| | | | Tinggal Peserta | tinggal peserta yang diinput dalam sistem BPJS Kesehatan pada saat melakukan pendaftaran (sesuai kode wilayah BPS) |
| 10. | <i>PSTV10</i> | category | Kabupaten/Kota | Tempat Tinggal Peserta Kabupaten/Kota tempat tinggal peserta yang diinput dalam sistem BPJS Kesehatan pada saat melakukan pendaftaran (sesuai kode wilayah BPS) |
| 11. | <i>PSTV11</i> | category | Kepemilikan faskes | Kepemilikan dari fasilitas kesehatan tempat peserta terdaftar |
| 12. | <i>PSTV12</i> | category | Jenis faskes | Jenis fasilitas kesehatan tempat peserta terdaftar |
| 13. | <i>PSTV13</i> | category | Provinsi Fasilitas Kesehatan | Peserta Terdaftar Provinsi faskes tempat peserta terdaftar (sesuai kode wilayah BPS) |
| 14. | <i>PSTV14</i> | category | Kabupaten/Kota Fasilitas Kesehatan Peserta Terdaftar | Kabupaten/Kota Fasilitas Kesehatan Peserta Terdaftar Kabupaten/Kota faskes tempat peserta terdaftar (sesuai kode wilayah BPS) |
| 15. | <i>PSTV15</i> | float32 | Bobot | Faktor pengali yang |

| | | | | |
|-----|---------------|----------|--------------------|--|
| | | | | menggambarkan jumlah individu di dalam populasi diwakili oleh individu di dalam sampel |
| 16. | <i>PSTV16</i> | int16 | Tahun Sampel | Tahun sampel diambil sebagai data kepesertaan |
| 17. | <i>PSTV17</i> | category | Status Kepesertaan | Status kepesertaan berdasarkan sistem BPJS Kesehatan per tanggal 31 Desember 2020 |
| 18. | <i>PSTV18</i> | float64 | Tahun meninggal | Tahun meninggal peserta yang tercatat pada sistem BPJS Kesehatan pada status kepesertaan meninggal |

3. TB2020_kepesertaan.dta

Tabel 4 Data Kepesertaan 2020

| No | Atribut | Tipe Atribut | Label Atribut | Keterangan |
|----|---------|-----------------|-----------------------|---|
| 1. | PSTV01 | int32 | Nomor peserta | Nomor identifikasi peserta yang bersifat unik dan |
| 2 | PSTV02 | int32 | Nomor keluarga | Nomor yang mengidentifikasi kepala keluarga dalam sampel dan berfungsi sebagai penanda keluarga (peserta BPJS Kesehatan dalam satu keluarga memiliki nomor kepala keluarga yang sama) |
| 3. | PSTV03 | datetime64 [ns] | Tanggal lahir peserta | Tanggal lahir peserta yang tercatat pada sistem BPJS Kesehatan ketika melakukan pendaftaran |
| 4. | PSTV04 | category | Hubungan Keluarga | Hubungan atau status peserta di dalam keluarga |
| 5. | PSTV05 | category | Jenis Kelamin | Jenis kelamin peserta |
| 6. | PSTV06 | category | Status perkawinan | Status perkawinan |
| 7. | PSTV07 | category | Kelas rawat | Kelas rawat peserta yang tercatat dalam sistem BPJS Kesehatan per tanggal 31 Desember 2020 |
| 8. | PSTV08 | category | Segmentasi peserta | Segmentasi peserta yang tercatat pada sistem |

| | | | | |
|-----|--------|----------|--|--|
| | | | | BPJS Kesehatan per tanggal 31 Desember 2020 |
| 9. | PSTV09 | category | Provinsi Tempat Tinggal Peserta | Provinsi tempat tinggal peserta yang diinput dalam sistem BPJS Kesehatan pada saat melakukan pendaftaran (sesuai kode wilayah BPS) |
| 10. | PSTV10 | category | Kabupaten/Kota Tempat Tinggal Peserta | Kabupaten/Kota tempat tinggal peserta yang diinput dalam sistem BPJS Kesehatan pada saat melakukan pendaftaran (sesuai kode wilayah BPS) |
| 11. | PSTV11 | category | Kepemilikan faskes | Kepemilikan dari fasilitas kesehatan tempat peserta terdaftar |
| 12. | PSTV12 | category | Jenis faskes | Jenis fasilitas kesehatan tempat peserta terdaftar |
| 13. | PSTV13 | category | Provinsi Fasilitas Kesehatan Peserta Terdaftar | Provinsi faskes tempat peserta terdaftar (sesuai kode wilayah BPS) |
| 14. | PSTV14 | category | Kabupaten/Kota Fasilitas Kesehatan Peserta Terdaftar | Kabupaten/Kota faskes tempat peserta terdaftar (sesuai kode wilayah BPS) |
| 15. | PSTV15 | float32 | Bobot | Faktor pengali yang menggambarkan jumlah individu di dalam populasi diwakili oleh individu di dalam sampel |

| | | | | |
|-----|--------|----------|-----------------------|---|
| 16. | PSTV16 | int16 | Tahun Sampel | Tahun sampel diambil sebagai data kepesertaan |
| 17. | PSTV17 | category | Status Kepesertaan | Status kepesertaan berdasarkan sistem BPJS Kesehatan per tanggal 31 Desember 2020 |
| 18. | PSTV18 | float64 | Tahun meninggal | Tahun meninggal peserta yang tercatat pada sistem BPJS Kesehatan pada status kepesertaan meninggal. |

4. TB2021_kepesertaan.dta

Tabel 5 Data Kepesertaan 2021

| No | Atribut | Tipe Atribut | Label Atribut | Keterangan |
|----|---------|--------------|-----------------------|---|
| 1. | PSTV01 | int32 | Nomor peserta | Nomor identifikasi peserta yang bersifat unik dan |
| 2 | PSTV02 | int32 | Nomor keluarga | Nomor yang mengidentifikasi kepala keluarga dalam sampel dan berfungsi sebagai penanda keluarga (peserta BPJS Kesehatan dalam satu keluarga memiliki nomor kepala keluarga yang sama) |
| 3. | PSTV03 | datetime64 | Tanggal lahir peserta | Tanggal lahir peserta yang tercatat pada sistem BPJS Kesehatan ketika melakukan pendaftaran |
| 4. | PSTV04 | category | Hubungan | Hubungan atau status |

| | | | | |
|-----|--------|----------|---------------------------------------|--|
| | | | Keluarga | peserta di dalam keluarga |
| 5. | PSTV05 | category | Jenis Kelamin | Jenis kelamin peserta |
| 6. | PSTV06 | category | Status perkawinan | Status perkawinan |
| 7. | PSTV07 | category | Kelas rawat | Kelas rawat peserta yang tercatat dalam sistem BPJS Kesehatan per tanggal 31 Desember 2020 |
| 8. | PSTV08 | category | Segmentasi peserta | Segmentasi peserta yang tercatat pada sistem BPJS Kesehatan per tanggal 31 Desember 2020 |
| 9. | PSTV09 | category | Provinsi Tempat Tinggal Peserta | Provinsi tempat tinggal peserta yang diinput dalam sistem BPJS Kesehatan pada saat melakukan pendaftaran (sesuai kode wilayah BPS) |
| 10. | PSTV10 | category | Kabupaten/Kota Tempat Tinggal Peserta | Kabupaten/Kota tempat tinggal peserta yang diinput dalam sistem BPJS Kesehatan pada saat melakukan pendaftaran (sesuai kode wilayah BPS) |
| 11. | PSTV11 | category | Kepemilikan faskes | Kepemilikan dari fasilitas kesehatan tempat peserta terdaftar |
| 12. | PSTV12 | category | Jenis faskes | Jenis fasilitas kesehatan tempat peserta terdaftar |

| | | | | |
|-----|--------|----------|--|--|
| 13. | PSTV13 | category | Provinsi Fasilitas Kesehatan Peserta Terdaftar | Provinsi faskes tempat peserta terdaftar (sesuai kode wilayah BPS) |
| 14. | PSTV14 | category | Kabupaten/Kota Fasilitas Kesehatan Peserta Terdaftar | Kabupaten/Kota faskes tempat peserta terdaftar (sesuai kode wilayah BPS) |
| 15. | PSTV15 | float32 | Bobot | Faktor pengali yang menggambarkan jumlah individu di dalam populasi diwakili oleh individu di dalam sampel |
| 16. | PSTV16 | int16 | Tahun Sampel | Tahun sampel diambil sebagai data kepesertaan |
| 17. | PSTV17 | category | Status Kepesertaan | Status kepesertaan berdasarkan sistem BPJS Kesehatan per tanggal 31 Desember 2020 |
| 18. | PSTV18 | float64 | Tahun meninggal | Tahun meninggal peserta yang tercatat pada sistem BPJS Kesehatan pada status kepesertaan meninggal. |

2.2.1 Fungsi head()

Menampilkan data untuk mengetahui isi data dari dataset adalah proses penting untuk mengetahui data yang kita miliki. Adapun kode program yang digunakan untuk menampilkan data adalah sebagai berikut:

```
df.head()
```

Output:

1. TB20152021_fktptonkapitasi.dta

data_fktptonkapitasi = pd.read_stata('/content/drive/MyDrive/DAMI/TB20152021_fktptonkapitasi.dta')

data_fktptonkapitasi.head()

| | PSTW01 | PSTW02 | PSTV15 | PNK02 | PNK03 | PNK04 | PNK05 | PNK06 | PNK07 | PNK08 | ... | PNK10 | PNK11 | PNK12 | PNK13 | PNK13A | PNK14 | PNK15 | PNK16 | PNK17 | PNK18 |
|---|-----------|-----------|-----------|------------------|------------|------------|------------|--------------------|----------|--------------------------|-----|-------------------|-------|----------|------------------------------------|--------|-------|----------------------------------|----------------------------------|--------|--------|
| 0 | 93858078 | 93216423 | 7.659537 | 183920215Y000376 | 2015-02-26 | 2015-02-26 | 2015-02-27 | SULAWESI SELATAN | BARRU | PEMERINTAH KABUPATENKOTA | — | RAIWA INAP | RITP | PBI APBN | NaN | 9999 | 9999 | Evakuasi medis / Ambulans Darat | 120000 | 120000 | |
| 1 | 93747649 | 346217457 | 1.800196 | 19500915Y000074 | 2015-09-21 | 2015-09-23 | 2015-09-23 | KALIMANTAN SELATAN | TAPIN | PEMERINTAH KABUPATENKOTA | — | RAIWA INAP | RITP | PBI APBN | NaN | 9999 | 9999 | Rawat Inap di R. Perawatan Biasa | 240000 | 240000 | |
| 2 | 359887820 | 72989971 | 0.959155 | 250630919P000299 | 2019-09-26 | 2019-09-29 | 2019-09-29 | PAPUA | JAYAPURA | TNI AD | — | KLINIK RAIWA INAP | RITP | PPU | A01 Typhoid and paratyphoid fevers | A01 | A01 | Typhoid fever | Rawat Inap di R. Perawatan Biasa | 600000 | 600000 |
| 3 | 84126594 | 84126594 | 1.110887 | 326360919P001086 | 2019-09-09 | 2019-09-12 | 2019-09-12 | JAWA TENGAH | REMBANG | PEMERINTAH KABUPATENKOTA | — | RAIWA INAP | RITP | PBI APBN | A01 Typhoid and paratyphoid fevers | A01 | A01 | Typhoid and paratyphoid fevers | Rawat Inap di R. Perawatan Biasa | 480000 | 480000 |
| 4 | 87558937 | 62126532 | 32.451832 | 252721019P001142 | 2019-10-14 | 2019-10-17 | 2019-10-17 | JAWA TENGAH | KEBUMEN | PEMERINTAH KABUPATENKOTA | — | RAIWA INAP | RITP | PBI APBN | A01 Typhoid and paratyphoid fevers | A01 | A01 | Typhoid and paratyphoid fevers | Rawat Inap di R. Perawatan Biasa | 600000 | 600000 |

K rows x 21 columns

2. TB2019_kepesertaan.dta

```
[7] data_kepesertaan2019 = pd.read_stata('/content/drive/MyDrive/DAMI/TB2019_kepesertaan.dta')
data_kepesertaan2019.head()
```

| | PSTW01 | PSTW02 | PSTW03 | PSTW04 | PSTW05 | PSTW06 | PSTW07 | PSTW08 | PSTW09 | PSTW10 | PSTW11 | PSTW12 | PSTW13 | PSTW14 | PSTW15 | PSTW16 | PSTW17 | PSTW18 |
|---|-----------|-----------|------------|---------|-----------|-------------|-----------|----------|--------|---------------|--------------------------|-----------|--------|---------------|----------|--------|--------|--------|
| 0 | 21611150 | 21611150 | 1957-09-12 | PESERTA | PEREMPUAN | KAWIN | KELAS I | PPU | ACEH | ACEH BESAR | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH BESAR | 1.157796 | 2019 | AKTIF | NaN |
| 1 | 94343049 | 96772919 | 1961-12-03 | SUAMI | LAGI-LAKI | KAWIN | KELAS I | PPU | ACEH | ACEH TENGGARA | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH TENGGARA | 1.556921 | 2019 | AKTIF | NaN |
| 2 | 83393824 | 298548714 | 2002-10-05 | ANAK | LAGI-LAKI | BELUM KAWIN | KELAS I | PPU | ACEH | ACEH TENGGARA | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH TENGGARA | 1.159913 | 2019 | AKTIF | NaN |
| 3 | 328637885 | 328537885 | 1989-07-13 | PESERTA | PEREMPUAN | BELUM KAWIN | KELAS III | PBI APBD | ACEH | ACEH TAMANG | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH TAMANG | 9.436164 | 2019 | AKTIF | NaN |
| 4 | 67805935 | 67805935 | 1972-11-13 | PESERTA | LAGI-LAKI | KAWIN | KELAS I | PPU | ACEH | ACEH TAMANG | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH TAMANG | 0.899331 | 2019 | AKTIF | NaN |

3. TB2020_kepesertaan.dta

```
data_kepesertaan2020 = pd.read_stata('/content/drive/MyDrive/DAMI/TB2020_kepesertaan.dta')
data_kepesertaan2020.head()
```

| | PSTW01 | PSTW02 | PSTW03 | PSTW04 | PSTW05 | PSTW06 | PSTW07 | PSTW08 | PSTW09 | PSTW10 | PSTW11 | PSTW12 | PSTW13 | PSTW14 | PSTW15 | PSTW16 | PSTW17 | PSTW18 |
|---|-----------|-----------|------------|---------|-----------|-------------|----------|---------------|----------------|--------------|--------------------------|----------------|----------------|--------------|-----------|--------|-------------|--------|
| 0 | 32853965 | 31945523 | 1958-11-27 | SUAMI | LAGI-LAKI | KAWIN | KELAS I | PPU | ACEH | ACEH TENGAH | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH TENGAH | 1.531137 | 2020 | MENINGGAL | 2020.0 |
| 1 | 8555967 | 8555967 | 1936-07-01 | PESERTA | LAGI-LAKI | KAWIN | KELAS I | BUKAN PEKERJA | ACEH | ACEH TENGAH | POLRI | KLINIK PRATAMA | ACEH | BIREUEN | 0.998777 | 2020 | AKTIF | NaN |
| 2 | 8797049 | 249479 | 1950-07-21 | SUAMI | LAGI-LAKI | KAWIN | KELAS I | BUKAN PEKERJA | ACEH | ACEH SELATAN | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH SELATAN | 30.096481 | 2020 | TIDAK AKTIF | NaN |
| 3 | 83472658 | 54408670 | 2012-08-12 | ANAK | PEREMPUAN | BELUM KAWIN | KELAS I | PPU | ACEH | ACEH BARAT | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH BARAT | 9.837102 | 2020 | AKTIF | NaN |
| 4 | 425874326 | 296852327 | 2004-12-30 | ANAK | LAGI-LAKI | BELUM KAWIN | KELAS II | PPU | SUMATERA UTARA | TOBA SAMOSIR | PEMERINTAH KABUPATENKOTA | PUSKESMAS | SUMATERA UTARA | TOBA SAMOSIR | 1.038886 | 2020 | AKTIF | NaN |

4. TB2021_kepesertaan.dta

```
data_kepesertaan2021 = pd.read_stata('/content/drive/MyDrive/DAMI/TB2021_kepesertaan.dta')
data_kepesertaan2021.head()
```

| | PSTW01 | PSTW02 | PSTW03 | PSTW04 | PSTW05 | PSTW06 | PSTW07 | PSTW08 | PSTW09 | PSTW10 | PSTW11 | PSTW12 | PSTW13 | PSTW14 | PSTW15 | PSTW16 | PSTW17 | PSTW18 |
|---|-----------|-----------|------------|---------|-----------|-------------|----------|---------------|----------------|--------------|--------------------------|----------------|----------------|--------------|-----------|--------|-------------|--------|
| 0 | 32853965 | 31945523 | 1958-11-27 | SUAMI | LAGI-LAKI | KAWIN | KELAS I | PPU | ACEH | ACEH TENGAH | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH TENGAH | 1.531137 | 2021 | MENINGGAL | 2020.0 |
| 1 | 8555967 | 8555967 | 1936-07-01 | PESERTA | LAGI-LAKI | KAWIN | KELAS I | BUKAN PEKERJA | ACEH | ACEH TENGAH | POLRI | KLINIK PRATAMA | ACEH | BIREUEN | 0.998777 | 2021 | AKTIF | NaN |
| 2 | 8797049 | 249479 | 1950-07-21 | SUAMI | LAGI-LAKI | KAWIN | KELAS I | BUKAN PEKERJA | ACEH | ACEH SELATAN | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH SELATAN | 30.096481 | 2021 | TIDAK AKTIF | NaN |
| 3 | 83472658 | 54408670 | 2012-08-12 | ANAK | PEREMPUAN | BELUM KAWIN | KELAS I | PPU | ACEH | ACEH BARAT | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH BARAT | 9.837102 | 2021 | AKTIF | NaN |
| 4 | 425874326 | 296852327 | 2004-12-30 | ANAK | LAGI-LAKI | BELUM KAWIN | KELAS II | PPU | SUMATERA UTARA | TOBA SAMOSIR | PEMERINTAH KABUPATENKOTA | PUSKESMAS | SUMATERA UTARA | TOBA SAMOSIR | 1.038886 | 2021 | AKTIF | NaN |

2.2.2 Fungsi shape()

Untuk mengetahui jumlah *row* dan *attribute* dari ke 6 data yang digunakan, dapat menggunakan kode program sebagai berikut:

```
df.shape
```

Output:

1. TB20152021_fktpnonkapitasi.dta

```
data_fkrtlnonkapasitas.shape
```

```
(36493, 21)
```

2. TB2019_kepesertaan.dta

```
data_kepesertaan2019.shape
```

```
(94966, 18)
```

3. TB2020_kepesertaan.dta

```
data_kepesertaan2020.shape
```

```
(94966, 18)
```

4. TB2021_kepesertaan.dta

```
data_kepesertaan2021.shape
```

```
(94966, 18)
```

2.2.3 Fungsi tail()

Untuk menampilkan data terakhir dari masing-masing data yang digunakan, dapat menggunakan kode program sebagai berikut:

```
df.tail()
```

Output:

1. TB20152021_fktpnonkapitasi.dta

| | PSTV01 | PSTV02 | PSTV15 | PNK02 | PNK03 | PNK04 | PNK05 | PNK06 | PNK07 | PNK08 | ... | PNK10 | PNK11 | PNK12 | PNK13 | PNK13A | PNK14 | PNK15 | PNK16 | PR |
|-------|-----------|-----------|-----------|------------------|------------|------------|------------|------------------|------------|---------------------------|-----|----------------|-------|----------|-------|---|-------|-------|---|-----|
| 36488 | 107435122 | 107435122 | 17.946516 | 108680818Y000770 | 2018-08-06 | 2018-08-06 | 2018-08-06 | SULAWESI SELATAN | WIAJO | PEMERINTAH KABUPATEN/KOTA | ... | RAWAT INAP | RJTP | PBI APBN | Z97 | Presence of (intrauterine) contraceptive device | Z97 | Z97 | Pelayanan KB : Pemasangan IUD / implant | 10X |
| 36489 | 447273237 | 435380768 | 2.330930 | 122070319Y000676 | 2019-03-09 | 2019-03-09 | 2019-03-09 | JAWA TIMUR | TUBAN | PEMERINTAH KABUPATEN/KOTA | ... | RAWAT INAP | RJTP | PBI APBD | Z97 | Presence of (intrauterine) contraceptive device | Z97 | Z97 | Pelayanan KB : Pemasangan IUD / implant | 10X |
| 36490 | 80666271 | 426470275 | 1.154869 | 296600118Y000277 | 2018-01-20 | 2018-01-20 | 2018-01-20 | PAPUA BARAT | RAJA AMPAT | PEMERINTAH KABUPATEN/KOTA | ... | NON RAWAT INAP | RJTP | PBI APBD | Z97 | Presence of (intrauterine) contraceptive device | Z97 | Z97 | Pelayanan KB : Suntik | 1t |
| 36491 | 93887676 | 93887676 | 3.204735 | 296600517Y000031 | 2017-05-03 | 2017-05-03 | 2017-05-03 | PAPUA BARAT | RAJA AMPAT | PEMERINTAH KABUPATEN/KOTA | ... | NON RAWAT INAP | RJTP | PBI APBD | Z97 | Presence of (intrauterine) contraceptive device | Z97 | Z97 | Pelayanan KB : Suntik | 1t |
| 36492 | 93887676 | 93887676 | 3.204735 | 296600818Y000004 | 2016-08-01 | 2016-08-01 | 2016-08-01 | PAPUA BARAT | RAJA AMPAT | PEMERINTAH KABUPATEN/KOTA | ... | NON RAWAT INAP | RJTP | PBI APBD | Z97 | Presence of (intrauterine) contraceptive device | Z97 | Z97 | Pelayanan KB : Suntik | 1t |

2. TB2019_kepesertaan.dta

| | PSTV01 | PSTV02 | PSTV03 | PSTV04 | PSTV05 | PSTV06 | PSTV07 | PSTV08 | PSTV09 | PSTV10 | PSTV11 | PSTV12 | PSTV13 | PSTV14 | PSTV15 | PSTV16 | PSTV17 |
|-------|-----------|-----------|------------|---------|-----------|-------------|-----------|--------|---------------------|-----------------|---------------------------|-------------|---------------------|-----------------|-----------|--------|-------------|
| 94961 | 446945417 | 446945417 | 1984-12-31 | PESERTA | LAKI-LAKI | KAWIN | KELAS III | PBPU | NUSA TENGGARA BARAT | LOMBOK BARAT | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | NUSA TENGGARA BARAT | LOMBOK BARAT | 3.706050 | 2019 | AKTIF |
| 94962 | 339605943 | 286316180 | 1989-03-17 | ISTRI | PEREMPUAN | KAWIN | KELAS III | PBPU | JAWA BARAT | SUKABUMI | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | JAWA BARAT | SUKABUMI | 35.093258 | 2019 | TIDAK AKTIF |
| 94963 | 292612825 | 301371045 | 1979-05-31 | ISTRI | PEREMPUAN | KAWIN | KELAS III | PBPU | SUMATERA SELATAN | EMPAT LAWANG | SWASTA | DOKTER UMUM | SUMATERA SELATAN | EMPAT LAWANG | 2.652437 | 2019 | AKTIF |
| 94964 | 412069355 | 412069355 | 1970-02-17 | PESERTA | LAKI-LAKI | KAWIN | KELAS III | PBPU | SUMATERA UTARA | TAPANULI TENGAH | SWASTA | DOKTER UMUM | SUMATERA UTARA | TAPANULI TENGAH | 14.152590 | 2019 | MENINGGAL |
| 94965 | 354440860 | 354440860 | 2000-09-23 | PESERTA | PEREMPUAN | BELUM KAWIN | KELAS II | PBPU | JAWA TENGAH | PEMALANG | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | JAWA TENGAH | PEMALANG | 1.224500 | 2019 | AKTIF |

3. TB2020_kepesertaan.dta

data_kepesertaan2020.tail()

| | PSTV01 | PSTV02 | PSTV03 | PSTV04 | PSTV05 | PSTV06 | PSTV07 | PSTV08 | PSTV09 | PSTV10 | PSTV11 | PSTV12 | PSTV13 | PSTV14 | PSTV15 | PSTV16 | PSTV17 |
|-------|-----------|-----------|------------|---------|-----------|-------------|-----------|----------|------------------|-------------------|---------------------------|-----------|------------------|-------------------|-----------|--------|-------------|
| 94961 | 441113640 | 353368323 | 1968-06-24 | ISTRI | PEREMPUAN | KAWIN | KELAS II | PBPU | BENGKULU | KOTA BENGKULU | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | BENGKULU | KOTA BENGKULU | 11.074059 | 2020 | MENINGGAL |
| 94962 | 348921684 | 348921684 | 1954-03-10 | PESERTA | PEREMPUAN | CERAI | KELAS III | PBI APBD | SUMATERA SELATAN | EMPAT LAWANG | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | SUMATERA SELATAN | EMPAT LAWANG | 1.168197 | 2020 | AKTIF |
| 94963 | 344293181 | 392640898 | 1975-07-07 | ISTRI | PEREMPUAN | KAWIN | KELAS III | PBI APBD | RIAU | BENGKALIS | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | RIAU | BENGKALIS | 1.783689 | 2020 | AKTIF |
| 94964 | 411717961 | 411717961 | 2010-03-13 | PESERTA | LAKI-LAKI | BELUM KAWIN | KELAS III | PBI APBD | PAPUA BARAT | MANOKWARI SELATAN | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | PAPUA BARAT | MANOKWARI SELATAN | 1.025151 | 2020 | AKTIF |
| 94965 | 278051593 | 278051593 | 1977-05-05 | PESERTA | LAKI-LAKI | KAWIN | KELAS III | PBPU | SUMATERA UTARA | SERDANG BEDAGAI | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | SUMATERA UTARA | SERDANG BEDAGAI | 14.769157 | 2020 | TIDAK AKTIF |

4. TB2021_kepesertaan.dta

data_kepesertaan2021.tail()

| | PSTV01 | PSTV02 | PSTV03 | PSTV04 | PSTV05 | PSTV06 | PSTV07 | PSTV08 | PSTV09 | PSTV10 | PSTV11 | PSTV12 | PSTV13 | PSTV14 | PSTV15 | PSTV16 | PSTV17 |
|-------|-----------|-----------|------------|---------|-----------|-------------|-----------|----------|------------------|-------------------|---------------------------|-----------|------------------|-------------------|-----------|--------|-------------|
| 94961 | 441113640 | 353368323 | 1968-06-24 | ISTRI | PEREMPUAN | KAWIN | KELAS II | PBPU | BENGKULU | KOTA BENGKULU | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | BENGKULU | KOTA BENGKULU | 11.074059 | 2021 | MENINGGAL |
| 94962 | 348921684 | 348921684 | 1954-03-10 | PESERTA | PEREMPUAN | CERAI | KELAS III | PBI APBD | SUMATERA SELATAN | EMPAT LAWANG | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | SUMATERA SELATAN | EMPAT LAWANG | 1.168197 | 2021 | AKTIF |
| 94963 | 344293181 | 392640898 | 1975-07-07 | ISTRI | PEREMPUAN | KAWIN | KELAS III | PBI APBD | RIAU | BENGKALIS | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | RIAU | BENGKALIS | 1.783689 | 2021 | AKTIF |
| 94964 | 411717961 | 411717961 | 2010-03-13 | PESERTA | LAKI-LAKI | BELUM KAWIN | KELAS III | PBI APBD | PAPUA BARAT | MANOKWARI SELATAN | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | PAPUA BARAT | MANOKWARI SELATAN | 1.025151 | 2021 | AKTIF |
| 94965 | 278051593 | 278051593 | 1977-05-05 | PESERTA | LAKI-LAKI | KAWIN | KELAS III | PBPU | SUMATERA UTARA | SERDANG BEDAGAI | PEMERINTAH KABUPATEN/KOTA | PUSKESMAS | SUMATERA UTARA | SERDANG BEDAGAI | 14.769157 | 2021 | TIDAK AKTIF |

2.2.4 Fungsi columns()

Untuk menampilkan data kolom dari masing-masing data yang digunakan, dapat menggunakan kode program sebagai berikut:

```
df.columns
```

Output :

1. TB20152021_fktpnnonkapitasi.dta

```
data_fkrtlnonkapasitas.columns
Index(['PSTV01', 'PSTV02', 'PSTV15', 'PNK02', 'PNK03', 'PNK04', 'PNK05',
      'PNK06', 'PNK07', 'PNK08', 'PNK09', 'PNK10', 'PNK11', 'PNK12', 'PNK13',
      'PNK13A', 'PNK14', 'PNK15', 'PNK16', 'PNK17', 'PNK18'],
      dtype='object')
```

2. TB2019_kepesertaan.dta

```
data_kepesertaan2019.columns  
  
Index(['PSTV01', 'PSTV02', 'PSTV03', 'PSTV04', 'PSTV05', 'PSTV06', 'PSTV07',  
      'PSTV08', 'PSTV09', 'PSTV10', 'PSTV11', 'PSTV12', 'PSTV13', 'PSTV14',  
      'PSTV15', 'PSTV16', 'PSTV17', 'PSTV18'],  
      dtype='object')
```

3. TB2020_kepesertaan.dta

```
34] data_kepesertaan2020.columns  
  
Index(['PSTV01', 'PSTV02', 'PSTV03', 'PSTV04', 'PSTV05', 'PSTV06', 'PSTV07',  
      'PSTV08', 'PSTV09', 'PSTV10', 'PSTV11', 'PSTV12', 'PSTV13', 'PSTV14',  
      'PSTV15', 'PSTV16', 'PSTV17', 'PSTV18'],  
      dtype='object')
```

4. TB2021_kepesertaan.dta

```
35] data_kepesertaan2021.columns  
  
Index(['PSTV01', 'PSTV02', 'PSTV03', 'PSTV04', 'PSTV05', 'PSTV06', 'PSTV07',  
      'PSTV08', 'PSTV09', 'PSTV10', 'PSTV11', 'PSTV12', 'PSTV13', 'PSTV14',  
      'PSTV15', 'PSTV16', 'PSTV17', 'PSTV18'],  
      dtype='object')
```

2.2.5 Fungsi info()

Untuk menampilkan gambaran mengenai masing-masing data yang digunakan, dapat menggunakan kode program sebagai berikut:

```
df.info()
```

Output :

1. TB20152021_fktpnonkapitasi.dta


```
data_fkrtlnonkapasitas.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
```

2. TB2019_kepesertaan.dta

```
data_kepesertaan2019.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
```

3. TB2020_kepesertaan.dta

```
data_kepesertaan2020.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
```

4. TB2021_kepesertaan.dta

```
data_kepesertaan2021.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
```

2.2.6 Fungsi dtypes()

Pada keenam data yang akan digunakan, terdapat beberapa atribut yang masing-masing memiliki tipe data yang berbeda. Adapun kode program yang digunakan untuk mengetahui

```
df.dtypes
```

tipe data dari dataset BPJS kesehatan 2022 adalah sebagai berikut :

Output:

1. TB20152021_fktpnonkapitasi.dta

```
data_fkrtlnonkapasitas.dtypes
```

```
PSTV01          int32
PSTV02          int32
PSTV15         float32
PNK02           object
PNK03    datetime64[ns]
PNK04    datetime64[ns]
PNK05    datetime64[ns]
PNK06           category
PNK07           category
PNK08           category
PNK09           category
PNK10           category
PNK11           category
PNK12           category
PNK13           category
PNK13A          object
PNK14          object
PNK15          object
PNK16           category
PNK17           int32
PNK18           int32
dtype: object
```

2. TB2019_kepesertaan.dta

```
data_kepesertaan2019.dtypes
```

```
PSTV01          int32
PSTV02          int32
PSTV03    datetime64[ns]
PSTV04           category
PSTV05           category
PSTV06           category
PSTV07           category
PSTV08           category
PSTV09           category
PSTV10           category
PSTV11           category
PSTV12           category
PSTV13           category
PSTV14           category
PSTV15         float32
PSTV16          int16
PSTV17           category
PSTV18         float64
dtype: object
```

3. TB2020_kepesertaan.dta

```
data_kepesertaan2020.dtypes
```

```
PSTV01          int32
PSTV02          int32
PSTV03    datetime64[ns]
PSTV04          category
PSTV05          category
PSTV06          category
PSTV07          category
PSTV08          category
PSTV09          category
PSTV10          category
PSTV11          category
PSTV12          category
PSTV13          category
PSTV14          category
PSTV15          float32
PSTV16          int16
PSTV17          category
PSTV18          float64
dtype: object
```

4. TB2021_kepesertaan.dta

```
] data_kepesertaan2021.dtypes
```

```
, PSTV01          int32
, PSTV02          int32
, PSTV03    datetime64[ns]
, PSTV04          category
, PSTV05          category
, PSTV06          category
, PSTV07          category
, PSTV08          category
, PSTV09          category
, PSTV10          category
, PSTV11          category
, PSTV12          category
, PSTV13          category
, PSTV14          category
, PSTV15          float32
, PSTV16          int16
, PSTV17          category
, PSTV18          float64
dtype: object
```

2.2.7 Fungsi describe()

Menampilkan data deskriptif statistik merupakan hal yang penting untuk mengetahui informasi statistik pada data Kontekstual TB di BPJS Kesehatan 2015-2022. Adapun kode program yang digunakan untuk mengetahui nilai deskriptif statistik pada keenam data yang dipilih yaitu:

```
df.describe()
```

Output:

1. TB20152021_fktpnonkapitasi.dta

```
data_fkrtlnonkapasitas.describe()
```

| | PSTV01 | PSTV02 | PSTV15 | PNK17 | PNK18 |
|-------|--------------|--------------|--------------|--------------|--------------|
| count | 3.649300e+04 | 3.649300e+04 | 36493.000000 | 3.649300e+04 | 3.649300e+04 |
| mean | 1.033491e+08 | 1.259502e+08 | 10.189317 | 1.353351e+05 | 1.353351e+05 |
| std | 1.224665e+08 | 1.390374e+08 | 21.183065 | 2.245414e+05 | 2.245414e+05 |
| min | 6.563800e+04 | 6.893000e+04 | 0.376170 | 0.000000e+00 | 0.000000e+00 |
| 25% | 3.120950e+07 | 3.389082e+07 | 1.516612 | 2.500000e+04 | 2.500000e+04 |
| 50% | 6.113467e+07 | 6.568918e+07 | 4.347620 | 4.500000e+04 | 4.500000e+04 |
| 75% | 8.974764e+07 | 9.855116e+07 | 10.225730 | 1.200000e+05 | 1.200000e+05 |
| max | 4.553364e+08 | 4.553826e+08 | 324.528839 | 5.625000e+06 | 5.625000e+06 |

2. TB2019_kepesertaan.dta

```
data_kepesertaan2019.describe()
```

| | PSTV01 | PSTV02 | PSTV15 | PSTV16 | PSTV18 |
|-------|--------------|--------------|--------------|---------|--------|
| count | 9.496600e+04 | 9.496600e+04 | 94966.000000 | 94966.0 | 4476.0 |
| mean | 1.830281e+08 | 1.903269e+08 | 8.461007 | 2019.0 | 2019.0 |
| std | 1.597607e+08 | 1.607588e+08 | 19.109184 | 0.0 | 0.0 |
| min | 1.394000e+03 | 3.687000e+03 | 0.156804 | 2019.0 | 2019.0 |
| 25% | 4.366800e+07 | 4.563971e+07 | 1.154869 | 2019.0 | 2019.0 |
| 50% | 8.745586e+07 | 9.109761e+07 | 2.845975 | 2019.0 | 2019.0 |
| 75% | 3.455679e+08 | 3.518411e+08 | 7.727345 | 2019.0 | 2019.0 |
| max | 4.553809e+08 | 4.553826e+08 | 365.923492 | 2019.0 | 2019.0 |

3. TB2020_kepesertaan.dta

```
data_kepesertaan2020.describe()
```

| | PSTV01 | PSTV02 | PSTV15 | PSTV16 | PSTV18 |
|-------|--------------|--------------|--------------|---------|-------------|
| count | 9.496600e+04 | 9.496600e+04 | 94966.000000 | 94966.0 | 7147.000000 |
| mean | 1.830281e+08 | 1.903269e+08 | 8.461007 | 2020.0 | 2019.370365 |
| std | 1.597607e+08 | 1.607588e+08 | 19.109184 | 0.0 | 0.482936 |
| min | 1.394000e+03 | 3.687000e+03 | 0.156804 | 2020.0 | 2019.000000 |
| 25% | 4.366800e+07 | 4.563971e+07 | 1.154869 | 2020.0 | 2019.000000 |
| 50% | 8.745586e+07 | 9.109761e+07 | 2.845975 | 2020.0 | 2019.000000 |
| 75% | 3.455679e+08 | 3.518411e+08 | 7.727345 | 2020.0 | 2020.000000 |
| max | 4.553809e+08 | 4.553826e+08 | 365.923492 | 2020.0 | 2020.000000 |

4. TB2021_kepesertaan.dta

```
data_kepesertaan2021.describe()
```

| | PSTV01 | PSTV02 | PSTV15 | PSTV16 | PSTV18 |
|-------|--------------|--------------|--------------|---------|-------------|
| count | 9.496600e+04 | 9.496600e+04 | 94966.000000 | 94966.0 | 8654.000000 |
| mean | 1.830281e+08 | 1.903269e+08 | 8.461007 | 2021.0 | 2019.656344 |
| std | 1.597607e+08 | 1.607588e+08 | 19.109184 | 0.0 | 0.759391 |
| min | 1.394000e+03 | 3.687000e+03 | 0.156804 | 2021.0 | 2019.000000 |
| 25% | 4.366800e+07 | 4.563971e+07 | 1.154869 | 2021.0 | 2019.000000 |
| 50% | 8.745586e+07 | 9.109761e+07 | 2.845975 | 2021.0 | 2019.000000 |
| 75% | 3.455679e+08 | 3.518411e+08 | 7.727345 | 2021.0 | 2020.000000 |
| max | 4.553809e+08 | 4.553826e+08 | 365.923492 | 2021.0 | 2021.000000 |

2.3 Memvalidasi Data

Dalam fase ini, data yang akan digunakan divalidasi dengan memastikan kelengkapan data untuk mencegah kemungkinan kesalahan atau masalah input data seperti nilai yang hilang atau duplikat. Oleh karena itu, dilakukan pemeriksaan terhadap atribut utama yang akan digunakan dalam dataset.

2.3.1 Missing Value

Pada sub-bab ini menjelaskan *Missing Value* yang terdapat pada 6 dataset yang digunakan dari folder Kontekstual TB. *Missing value* merupakan data atau informasi pada dataset yang tidak tersedia atau bernilai 'NaN'. Missing value terjadi karena informasi untuk sesuatu tentang objek tidak diberikan, sulit dicari, atau memang informasi tersebut tidak ada. Untuk memeriksa *Missing value* pada dataset digunakan kode program berikut:

```
df.isna()
```

Output :

1. TB20152021_fktpnonkapitasi.dta

Pada data ini terdapat missing value yang dapat dilihat pada gambar di bawah ini.

```
data_fkrtlnonkapasitas.isnull().sum()
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
```

2. TB2019_kepesertaan.dta

Pada data ini terdapat missing value yang dapat dilihat pada gambar di bawah ini.

```
data_kepesertaan2019.isnull().sum()
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    90490
dtype: int64
```

3. TB2020_kepesertaan.dta

Pada data ini terdapat missing value yang dapat dilihat pada gambar di bawah ini.

```
data_kepesertaan2020.isnull().sum()
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    87819
dtype: int64
```

4. TB2021_kepesertaan.dta

Pada data ini terdapat missing value yang dapat dilihat pada gambar di bawah ini.

```
data_kepesertaan2021.isnull().sum()
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    86312
dtype: int64
```

2.3.2 Duplicate Values

Pada sub bab ini menjelaskan *Duplicate Values* yang terdapat pada 6 dataset yang digunakan dari folder Kontekstual TB. *Duplicate Values* adalah suatu informasi atau data pada dataset yang memiliki karakteristik atau nilai yang sama atau identik dengan informasi atau data yang satu dengan data yang lainnya. Untuk memeriksa *Duplicate Values* dapat digunakan kode program berikut :

```
duplicate = df.duplicated()  
print(duplicate.sum())  
df[duplicate]
```

Output :

1. TB20152021_fktpnonkapitasi.dta

Pada data ini terdapat duplicate value yang dapat dilihat pada gambar di bawah ini.

```
▶ duplicate_fkrtlnonkapasitas = data_fkrtlnonkapasitas.duplicated()  
  duplicate_fkrtlnonkapasitas.sum()  
⇒ 487
```

2. TB2019_kepesertaan.dta

Pada data ini tidak terdapat duplicate value yang dapat dilihat pada gambar di bawah ini.

```
[71] duplicate_kepesertaan2019 = data_kepesertaan2019.duplicated()  
     duplicate_kepesertaan2019.sum()  
0
```

3. TB2020_kepesertaan.dta

Pada data ini tidak terdapat duplicate value yang dapat dilihat pada gambar di bawah ini.

```
▶ duplicate_kepesertaan2020 = data_kepesertaan2020.duplicated()  
  duplicate_kepesertaan2020.sum()  
⇒ 0
```

4. TB2021_kepesertaan.dta

Pada data ini tidak terdapat duplicate value yang dapat dilihat pada gambar di bawah ini.

```
[73] duplicate_kepesertaan2021 = data_kepesertaan2021.duplicated()  
     duplicate_kepesertaan2021.sum()  
0
```


BAB III

DATA PREPARATION

Data Preparation dilakukan untuk menghasilkan data yang memiliki kualitas tinggi. Berdasarkan penjelasan data pada bab 2, persiapan data dilakukan melalui beberapa tahapan, termasuk pembersihan data, integrasi data, transformasi data, dan reduksi data. Persiapan data merupakan langkah setelah pengumpulan data awal yang telah dilakukan dalam fase CRISP-DM sebelumnya, yaitu pemahaman bisnis, kemudian diikuti oleh proses menyiapkan data, pemilihan variabel yang akan dianalisis, dan pembersihan data. Pemrograman dilakukan menggunakan perangkat lunak pengolah data Google Collab.

3.1 Mengintegrasikan Data

Mengintegrasikan data diterapkan untuk menggabungkan data dari berbagai sumber ke penyimpanan data yang koheren seperti gudang data (data warehouse). Terdapat 4 data dengan format dta yang diintegrasikan dan menghasilkan satu dataset. Namun pertama sekali, data yang di merged adalah data kepesertaan dari tahun 2019-2021.

```
m1 = [data_kepesertaan2019, data_kepesertaan2020,
data_kepesertaan2021]
```

```
data_merge = pd.concat(m1)
data_merge
```

Output:

| | PSTV01 | PSTV02 | PSTV03 | PSTV04 | PSTV05 | PSTV06 | PSTV07 | PSTV08 | PSTV09 | PSTV10 | PSTV11 | PSTV12 | PSTV13 | PSTV14 | PSTV15 | PSTV16 | PSTV17 | PSTV18 |
|-------|-----------|-----------|------------|---------|-----------|-------------|-----------|----------|------------------|-------------------|--------------------------|-----------|------------------|-------------------|-----------|--------|-------------|--------|
| 0 | 21611150 | 21611150 | 1957-09-12 | PESERTA | PEREMPUAN | KAWIN | KELAS I | PPU | ACEH | ACEH BESAR | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH BESAR | 1.157796 | 2019 | AKTIF | NaN |
| 1 | 94343049 | 96772919 | 1961-12-03 | SUAMI | LAKI-LAKI | KAWIN | KELAS I | PPU | ACEH | ACEH TENGGERA | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH TENGGERA | 1.556821 | 2019 | AKTIF | NaN |
| 2 | 83393824 | 298548714 | 2002-10-05 | ANAK | LAKI-LAKI | BELUM KAWIN | KELAS I | PPU | ACEH | ACEH TENGGERA | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH TENGGERA | 1.159913 | 2019 | AKTIF | NaN |
| 3 | 328537885 | 328537885 | 1989-07-13 | PESERTA | PEREMPUAN | BELUM KAWIN | KELAS III | PBI APBD | ACEH | ACEH TAMIANG | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH TAMIANG | 9.436164 | 2019 | AKTIF | NaN |
| 4 | 67805935 | 67805935 | 1972-11-13 | PESERTA | LAKI-LAKI | KAWIN | KELAS I | PPU | ACEH | ACEH TAMIANG | PEMERINTAH KABUPATENKOTA | PUSKESMAS | ACEH | ACEH TAMIANG | 0.899331 | 2019 | AKTIF | NaN |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 94961 | 441113640 | 353368323 | 1968-06-24 | ISTRI | PEREMPUAN | KAWIN | KELAS II | PBP | BENGKULU | KOTA BENGKULU | PEMERINTAH KABUPATENKOTA | PUSKESMAS | BENGKULU | KOTA BENGKULU | 11.074059 | 2021 | MENINGGAL | 2019.0 |
| 94962 | 348921684 | 348921684 | 1954-03-10 | PESERTA | PEREMPUAN | CERAI | KELAS III | PBI APBD | SUMATERA SELATAN | EMPAT LAWANG | PEMERINTAH KABUPATENKOTA | PUSKESMAS | SUMATERA SELATAN | EMPAT LAWANG | 1.168197 | 2021 | AKTIF | NaN |
| 94963 | 344293181 | 392640898 | 1975-07-07 | ISTRI | PEREMPUAN | KAWIN | KELAS III | PBI APBD | RIAU | BENGKALIS | PEMERINTAH KABUPATENKOTA | PUSKESMAS | RIAU | BENGKALIS | 1.763689 | 2021 | AKTIF | NaN |
| 94964 | 411717961 | 411717961 | 2010-03-13 | PESERTA | LAKI-LAKI | BELUM KAWIN | KELAS III | PBI APBD | PAPUA BARAT | MANOKWARI SELATAN | PEMERINTAH KABUPATENKOTA | PUSKESMAS | PAPUA BARAT | MANOKWARI SELATAN | 1.025151 | 2021 | AKTIF | NaN |
| 94965 | 278051593 | 278051593 | 1977-05-05 | PESERTA | LAKI-LAKI | KAWIN | KELAS III | PBP | SUMATERA UTARA | SERDANG BEDAGAI | PEMERINTAH KABUPATENKOTA | PUSKESMAS | SUMATERA UTARA | SERDANG BEDAGAI | 14.769157 | 2021 | TIDAK AKTIF | NaN |

264898 rows x 18 columns

Kemudian hasil merge data kepesertaan digabungkan lagi dengan data fktponkapasiti, namun dikarenakan pada data kepesertaan sudah terdapat 3 atribut ('PSTV01', 'PSTV02', 'PSTV15'), maka pada data fktponkapasiti tidak perlu dimasukkan 3 kolom atribut tersebut.

```
result_merge = pd.merge(data_merge, data_fktpnonkapasitas,
on=['PSTV01', 'PSTV02', 'PSTV15'], how='inner')
```

```
result_merge
```

Output:

| | PSTV01 | PSTV02 | PSTV03 | PSTV04 | PSTV05 | PSTV06 | PSTV07 | PSTV08 | PSTV09 | PSTV10 | ... | PNK10 | PNK11 | PNK12 | PNK13 | PNK13A | PNK14 | PNK15 | PNK16 | PNK17 | PNK18 |
|---|-----------|-----------|------------|---------|-----------|-------------|-----------|---------------|-------------|-------------|-----|------------|-------|---------------|---------------------------------|--------|-------|-----------------------------|--|--------|--------|
| 0 | 328537885 | 328537885 | 1989-07-13 | PESERTA | PEREMPUAN | BELUM KAWIN | KELAS III | PBI APBD | ACEH | ACEH TAMANG | ... | RAWAT INAP | RITP | PBI APBD | O60 Single spontaneous delivery | O60 | O600 | Spontaneous vertex delivery | Paket Persalinan per Vaginam normal (oleh Bidan) | 700000 | 700000 |
| 1 | 328537885 | 328537885 | 1989-07-13 | PESERTA | PEREMPUAN | BELUM KAWIN | KELAS III | PBI APBD | ACEH | ACEH TAMANG | ... | RAWAT INAP | RITP | PBI APBD | O60 Single spontaneous delivery | O60 | O600 | Spontaneous vertex delivery | Paket Persalinan per Vaginam normal (oleh Bidan) | 700000 | 700000 |
| 2 | 328537885 | 328537885 | 1989-07-13 | PESERTA | PEREMPUAN | KAWIN | KELAS III | PBI APBD | ACEH | ACEH TAMANG | ... | RAWAT INAP | RITP | PBI APBD | O60 Single spontaneous delivery | O60 | O600 | Spontaneous vertex delivery | Paket Persalinan per Vaginam normal (oleh Bidan) | 700000 | 700000 |
| 3 | 75453396 | 75453396 | 1959-08-25 | PESERTA | LAKI-LAKI | KAWIN | KELAS I | PPU | JAWA TENGAH | BLORA | ... | RAWAT INAP | RJTP | BUKAN PEKERJA | R06 Abnormalities of breathing | R06 | R060 | Dyspnoea | Evaluasi medis / Ambulans Darat | 161700 | 161700 |
| 4 | 75453396 | 75453396 | 1959-08-25 | PESERTA | LAKI-LAKI | KAWIN | KELAS I | BUKAN PEKERJA | JAWA TENGAH | BLORA | ... | RAWAT INAP | RJTP | BUKAN PEKERJA | R06 Abnormalities of breathing | R06 | R060 | Dyspnoea | Evaluasi medis / Ambulans Darat | 161700 | 161700 |

Total hasil merge data kepesertaan 2019-2021 dan fktpnonkapasiti yaitu 109479 rows \times 36 columns.

3.2 Data Selection

Dalam hal ini akan dipilih data-data yang diperlukan. Maka dari itu, dilakukan *data.drop()* untuk menghapus feature yang tidak digunakan dalam pengerjaan proyek. Sesuai pembahasan sebelumnya, maka dilakukan drop feature. Pada tahap ini juga dilakukan penanganan terhadap data noise dengan cara mengecek data yang error, outlier, atau inconsistent. Adapun proses yang dilakukan pada data cleaning adalah menghapus objek data yang nilai (missing value) dan mengeliminasi atribut yang tidak mempunyai korelasi. Sebelum menghapus objek data yang terdapat missing value, kita dapat melakukan pengecekan dengan menggunakan fungsi *isna()* dan akan mengembalikan nilai boolean true jika kondisi benar atau terdapat data kosong dan false jika kondisi data tidak mengandung nilai (missing value). Fungsi *sum()* dapat kita gunakan untuk mengetahui jumlah data yang missing value pada setiap atribut pada data. Kedua fungsi ini dapat kita hubungkan agar kita dapat melakukan pengecekan dan mendapat jumlah data yang missing.

3.3 Data Cleaning

3.3.1 Feature Selection

Pada sub bab ini menjelaskan Eliminasi data yang tidak relevan (inconsistent) yang terdapat pada dataset fraud detection train. Eliminasi data yang tidak relevan (inconsistent) dilakukan pada 17 atribut. Melakukan eliminasi dilakukan dengan fungsi drop() dengan menggunakan kode program berikut :

```
df.drop(['PSTV18', 'PSTV17', 'PSTV16', 'PSTV11', 'PSTV12', 'PSTV08',  
'PSTV13', 'PSTV07', 'PSTV04', 'PSTV06', 'PNK13A', 'PNK12', 'PSTV05',  
'PNK06', 'PNK09', 'PNK08', 'PNK10'], axis=1, inplace=True)
```

3.3.2 Missing Value

Pada sub bab ini menjelaskan Missing Value yang terdapat pada dataset fraud detection train. Missing value merupakan data atau informasi pada dataset yang tidak tersedia atau bernilai 'NaN'. Missing value terjadi karena informasi untuk sesuatu tentang objek tidak diberikan, sulit dicari, atau memang informasi tersebut tidak ada. Memeriksa Missing value pada dataset digunakan kode program berikut:

```
result_merge.isnull().sum()
```

Output:

```
┌ 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  35216  
┌ 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
```

```
# handling missing value
result_merge['PSTV18'] = result_merge['PSTV18'].fillna('Belum
Meninggal')
```

```
df['PSTV18']
```

Output:

```
0      Belum Meninggal
1      Belum Meninggal
2      Belum Meninggal
3      Belum Meninggal
4      Belum Meninggal
...
109474  Belum Meninggal
109475  Belum Meninggal
109476  Belum Meninggal
109477  Belum Meninggal
109478  Belum Meninggal
Name: PSTV18, Length: 109479, dtype: object
```

3.3.3 Data Duplicated

```
result_merge.duplicated().sum()
```

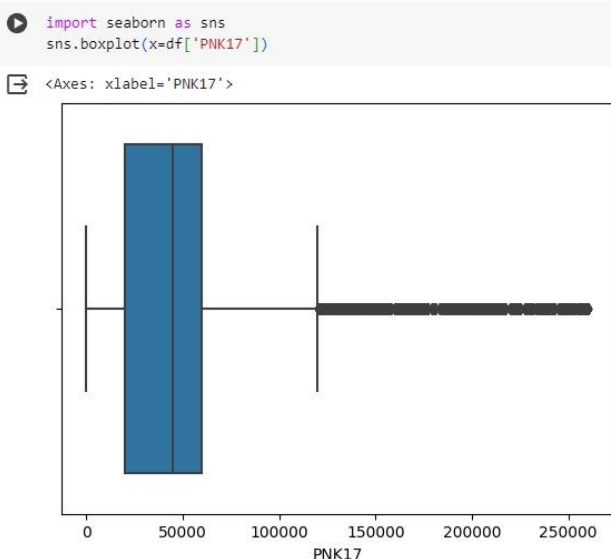
Output :

1461

3.3.4 Outlier

```
import seaborn as sns
sns.boxplot(x=df['PNK17'])
```

Output:



```

Q1 = df.quantile(0.25, numeric_only=True)
Q3 = df.quantile(0.75, numeric_only=True) # Corrected to 0.75 for
the third quartile
IQR = Q3 - Q1

df = df[~((df.lt(Q1 - 1.5 * IQR)) | (df.gt(Q3 + 1.5 *
IQR)))].any(axis=1)]

df.shape

```

Output:

```

] Q1 = df.quantile(0.25, numeric_only=True)
Q3 = df.quantile(0.75, numeric_only=True) # Corrected to 0.75 for the third quartile
IQR = Q3 - Q1

df = df[~((df.lt(Q1 - 1.5 * IQR)) | (df.gt(Q3 + 1.5 * IQR)))].any(axis=1)]

df.shape

(65643, 20)

```

3.3.5 Menampilkan proporsi kelas label

Menampilkan proporsi kelas label untuk mengetahui apakah dataset yang digunakan sudah cukup seimbang dilihat dari proporsi kelas label. Adapun kode program yang digunakan untuk menampilkan proporsi kelas label adalah sebagai berikut :

```

All = df.shape[0]
RITP = df[df['PNK11'] == 'RITP']
RJTP = df[df['PNK11'] == 'RJTP']
PROMOTIF = df[df['PNK11'] == 'PROMOTIF']

totalRITP = len(RITP)/All
totalRJTP = len(RJTP)/All
totalPromotif = len(PROMOTIF)/All

print('RITP :', totalRITP * 100, '%')
print('RJTP :', totalRJTP * 100, '%')
print('PROMOTIF :', totalPromotif * 100, '%')

```

Output:

```

RITP : 12.933987340037815 %
RJTP : 35.045077138081275 %
PROMOTIF : 52.02093552188091 %

```

```

from imblearn.under_sampling import RandomUnderSampler

train_data = pd.concat([X_train, Y_train], axis=1)
class_counts = Y_train['PNK11'].value_counts()
undersample_size = min(class_counts)
rus = RandomUnderSampler(sampling_strategy={0: undersample_size, 1:
undersample_size, 2: undersample_size}, random_state=42)
X_resampled, y_resampled = rus.fit_resample(X_train,
Y_train['PNK11'])
print(pd.Series(y_resampled).value_counts(normalize=True))

```

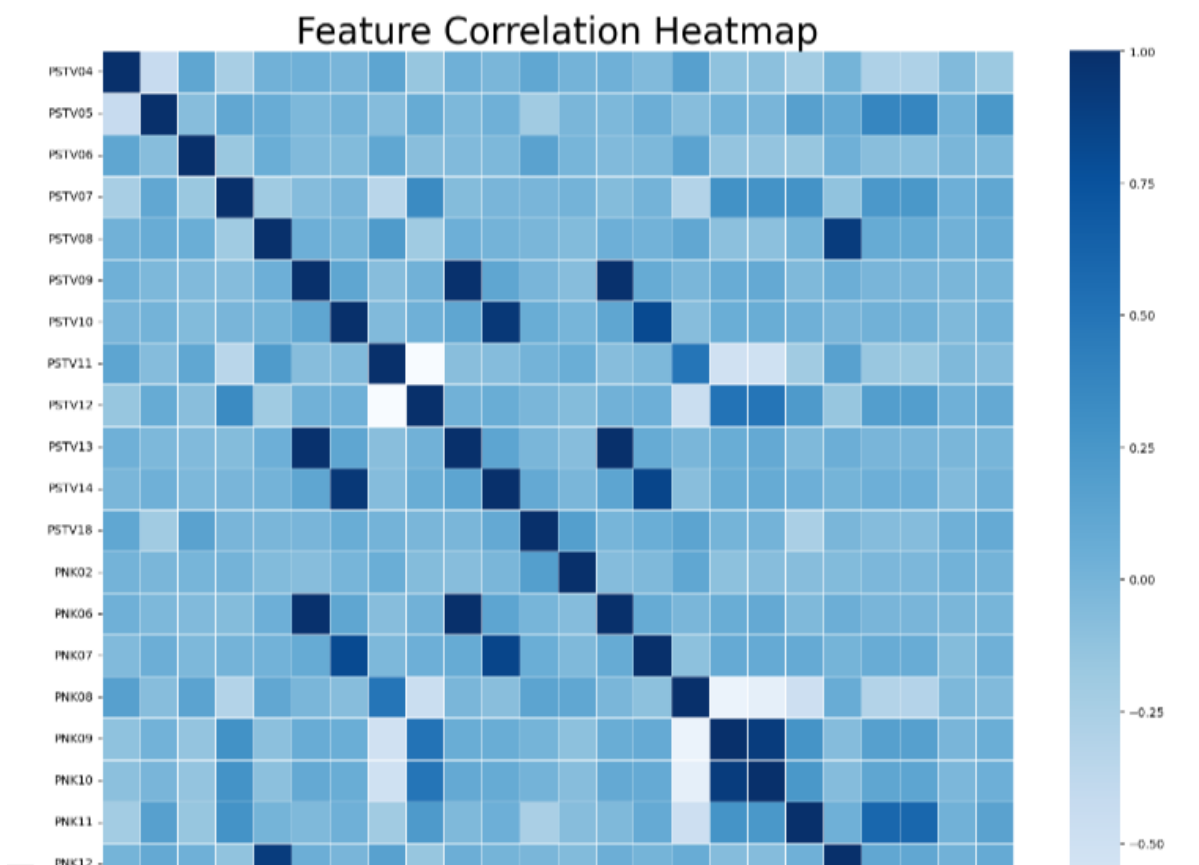
Output:

```

0    0.333333
1    0.333333
2    0.333333
Name: PNK11, dtype: float64

```

3.3.6 Feature Correlation



Gambar 1 Feature Correlation

3.4 Menentukan Label

Dataset yang digunakan sudah memiliki label data yang dapat langsung digunakan untuk pemodelan nantinya, sehingga tahapan pelabelan data tidak perlu dilakukan lagi.

3.5 Mengkonstruksi Data

Pada fase ini dilakukan konstruksi data dengan melakukan transformasi terhadap atribut.

3.5.1 Transformasi

Pada fase ini dilakukan transformasi data dengan melakukan transformasi data ke semua nilai yang bernilai object agar bernilai numerik.

```
df['PNK11'].replace(to_replace = ['RITP', 'RJTP', 'PROMOTIF'], value
= [2, 1, 0], inplace = True)
```

```
data_categorical = df.select_dtypes(include=['object']).copy()
```

```
data_categorical.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 109479 entries, 0 to 109478
Data columns (total 13 columns):
#   Column      Non-Null Count  Dtype
---  ---
0    PSTV03      109479 non-null  object
1    PSTV09      109479 non-null  object
2    PSTV10      109479 non-null  object
3    PSTV14      109479 non-null  object
4    PNK02       109479 non-null  object
5    PNK03       109479 non-null  object
6    PNK04       109479 non-null  object
7    PNK05       109479 non-null  object
8    PNK07       109479 non-null  object
9    PNK13       109473 non-null  object
10   PNK14       109479 non-null  object
11   PNK15       109479 non-null  object
12   PNK16       109479 non-null  object
dtypes: object(13)
memory usage: 10.9+ MB
```

```
from sklearn.preprocessing import LabelEncoder
LE = LabelEncoder()
```

```
df['PSTV03'] = LE.fit_transform(df['PSTV03'])
#df['PSTV04'] = LE.fit_transform(df['PSTV04'])
#df['PSTV05'] = LE.fit_transform(df['PSTV05'])
#df['PSTV06'] = LE.fit_transform(df['PSTV06'])
#df['PSTV12'] = LE.fit_transform(df['PSTV12'])
#df['PSTV07'] = LE.fit_transform(df['PSTV07'])
df['PSTV09'] = LE.fit_transform(df['PSTV09'])
df['PSTV10'] = LE.fit_transform(df['PSTV10'])
```

```

#df['PSTV11'] = LE.fit_transform(df['PSTV11'])
#df['PSTV13'] = LE.fit_transform(df['PSTV13'])
df['PSTV14'] = LE.fit_transform(df['PSTV14'])
#df['PSTV17'] = LE.fit_transform(df['PSTV17'])
#df['PSTV18'] = LE.fit_transform(df['PSTV18'])
df['PNK02'] = LE.fit_transform(df['PNK02'])
df['PNK03'] = LE.fit_transform(df['PNK03'])
df['PNK04'] = LE.fit_transform(df['PNK04'])
df['PNK05'] = LE.fit_transform(df['PNK05'])
#df['PNK06'] = LE.fit_transform(df['PNK06'])
df['PNK07'] = LE.fit_transform(df['PNK07'])
#df['PNK08'] = LE.fit_transform(df['PNK08'])
#df['PNK09'] = LE.fit_transform(df['PNK09'])
#df['PNK10'] = LE.fit_transform(df['PNK10'])
#df['PSTV08'] = LE.fit_transform(df['PSTV08'])
#df['PNK12'] = LE.fit_transform(df['PSTV12'])
df['PNK13'] = LE.fit_transform(df['PNK13'])
#df['PNK13A'] = LE.fit_transform(df['PNK13A'])
df['PNK14'] = LE.fit_transform(df['PNK14'])
df['PNK15'] = LE.fit_transform(df['PNK15'])
df['PNK16'] = LE.fit_transform(df['PNK16'])

```

```
df.info()
```

Output:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 109479 entries, 0 to 109478
Data columns (total 19 columns):
 #   Column  Non-Null Count  Dtype
---  -
 0   PSTV01  109479 non-null  int64
 1   PSTV02  109479 non-null  int64
 2   PSTV03  109479 non-null  int64
 3   PSTV09  109479 non-null  int64
 4   PSTV10  109479 non-null  int64
 5   PSTV14  109479 non-null  int64
 6   PSTV15  109479 non-null  float64
 7   PNK02   109479 non-null  int64
 8   PNK03   109479 non-null  int64
 9   PNK04   109479 non-null  int64
10  PNK05   109479 non-null  int64
11  PNK07   109479 non-null  int64
12  PNK11   109479 non-null  int64
13  PNK13   109479 non-null  int64
14  PNK14   109479 non-null  int64
15  PNK15   109479 non-null  int64
16  PNK16   109479 non-null  int64
17  PNK17   109479 non-null  int64
18  PNK18   109479 non-null  int64
dtypes: float64(1), int64(18)
memory usage: 15.9 MB

```


BAB IV

MODELLING

Pada bab ini akan dijelaskan mengenai pemilihan teknik modelling, test design, membangun model, dan melakukan penilaian terhadap model yang telah dibangun. Pada bab sebelumnya, kelompok telah mempersiapkan data yang digunakan untuk membangun model.

4.1 Membangun Skenario Pengujian

Berikut langkah-langkah yang dilakukan tim proyek meliputi:

- 1) Data Preparation
 - a) Memilah Data
 - b) Membersihkan Data
 - c) Mengkonstruksi Data
 - d) Mengintegrasikan Data
- 2) Modeling
 - a) Membangun Skenario Pengujianh
 - b) Membangun Model
- 3) Model Evaluasi
 - a) Membangun Hasil Pemodelan
 - b) Melakukan Review Proses Pemodelan

Model klasifikasi yang diharapkan berdasarkan yang dibangun memenuhi syarat berupa nilai dari Precision > 0.54, Accuracy > 0.56 dan Recall > 0.65.

4.2 Artificial Neural Network

Artificial Neural Network adalah salah satu metode Deep Learning yang bekerja seperti jaringan saraf manusia. Kategori ANN didasarkan pada metode supervised dan unsupervised. Bentuk sederhana dari ANN adalah persepsi yang terdiri dari satu dengan dua masukan dan satu keluaran. Persepsi digunakan untuk klasifikasi data yang diubah menjadi dua kelas terpisah.

Arsitektur dari ANN adalah sebagai berikut:

- a. Lapisan Input (Input Layer): Menerima nilai input
- b. Lapisan tersembunyi (Hidden Layer): Satu set neuron antara lapisan input dan output.
- c. Lapisan Output (Output layer): Memiliki satu neuron dan outputnya berkisar antara 0 dan 1 atau lebih besar dari 0 dan lebih kecil dari 1.

4.3 Build Model

Terdapat 3 informasi yang digunakan dalam pengambilan keputusan data mining, yaitu Parameter settings. Parameter settings merupakan parameter yang digunakan pada model untuk memberikan hasil yang baik model yang dihasilkan. Deskripsi dari hasil pemodelan, termasuk performance dan data issued yang terjadi selama pengeksekusian model dan eksplorasi hasil. Dalam pembuatan/pembangunan model ini, kami melakukan tahapan sebagai berikut:

1) Import Library

Langkah pertama adalah mengimport library yang dibutuhkan seperti numpy, pandas, seaborn, matplotlib, dan sklearn. Untuk penjelasan mengenai library tersebut telah dijelaskan di bab sebelumnya.

Kode:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
```

2) Import Dataset

```
df = pd.read_csv('/content/drive/MyDrive/DAMI/data_dami.csv')
```

3) Defenisi x dan y

```
X = df.drop("PSTV12", axis = 1)
Y = df['PSTV12']
```

4) Membagi data train dan data test

Dataset dibagi menjadi 70% training dataset dan 30% testing dataset menggunakan library sklearn.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size =
0.30)

print(f"Shape of X_train is: {X_train.shape}")
print(f"Shape of Y_train is: {Y_train.shape}\n")
print(f"Shape of X_test is: {X_test.shape}")
print(f"Shape of Y_test is: {Y_test.shape}")
```

5) Build Model

```
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
```

6) Compile Model

```
model.compile(optimizer = Adam(learning_rate = 0.001),  
loss = 'binary_crossentropy', metrics = ['accuracy'])
```

7) Train

```
result = model.fit(X_train, Y_train, epochs = 20, batch_size = 10)
```

BAB V

EVALUATION

Berdasarkan CRISP-DM tahap kelima yaitu tahap evaluasi terhadap model yang telah dirancang untuk melakukan fraud detection pada dataset BPJS menggunakan algoritma ANN. Tujuan dari evaluasi yaitu agar hasil yang didapatkan pada tahap modelling lebih maksimal dan sesuai dengan tujuan yang ingin dicapai pada tahap business understanding. Berikut merupakan hasil evaluasi model yang didapatkan

- Test Accuracy : 0.70 / 70%
- Test Precision : 76%
- Test Recall : 98%
- F1 Score : 85%

BAB VI

DEPLOYMENT

Tahap pembuatan model pada umumnya bukanlah akhir dari proyek. Meskipun tujuan model adalah untuk meningkatkan pengetahuan tentang data, pengetahuan yang diperoleh perlu disajikan sedemikian rupa sehingga user dapat menggunakannya dengan mudah. Tahap deployment yang dapat dilakukan biasanya dengan melakukan personalisasi pada halaman-halaman web. Pada proyek ini, model yang sebelumnya sudah dibangun dan sudah di evaluasi oleh tim selanjutnya dikembangkan menjadi sebuah sistem sederhana. Berikut tahapan yang dilakukan dalam men-deploy data mining model.

6.1 Mempersiapkan interface dan Script Code

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

Desain terdapat pada folder :

```
> static
> templates
```

Static terdiri dari beberapa folder yaitu css, images, js:

```
▼ static
  > css
  > img
  > js
```

Templates terdiri dari file index.html

```
▼ templates
  <> index.html
```

6.1.1 Membuat interface HTML

Dalam Classification of Tuberculosis Disease from 2019-2021 using Artificial Neural Network (ANN), yang pertama sekali dilakukan adalah pengumpulan data dan kemudian menerapkan Artificial Neural Network untuk memprediksi hasil klasifikasi. Oleh karena itu, dalam pengumpulan data tim kerja membuat formulir html untuk 5 kolom. Berikut tampilan code index.html dan layout nya.

```

<!-- System Section start -->
<section id="system" class="system">
  <h2>BPJS Classification <span>using ANN</span></h2>
  <p>
    Fill-up the form to predict
  </p>

  <form action="/predict" method="post">
    <div class="input-group">
      <input type="text" id="pstv10" name="pstv10" required="required" placeholder="Nomor Peserta *"
        onFocus="if (this.value == 'Nomor Peserta') this.value = '';"
        onBlur="if (this.value == '') this.value = 'Nomor Peserta';" />
    </div>
    <div class="input-group">
      <input type="text" id="pstv10" name="pstv10" required="required" placeholder="Kabupaten/Kota Tempat Tinggal Peserta *"
        onFocus="if (this.value == 'Kabupaten/Kota Tempat Tinggal Peserta ') this.value = '';"
        onBlur="if (this.value == '') this.value = 'Kabupaten/Kota Tempat Tinggal Peserta ';" />
    </div>
    <div class="input-group">
      <input type="text" id="pstv14" name="pstv14" required="required" placeholder="Kabupaten/Kota Fasilitas Kesehatan Peserta Terdaftar *"
        onFocus="if (this.value == 'Kabupaten/Kota Fasilitas Kesehatan Peserta Terdaftar') this.value = '';"
        onBlur="if (this.value == '') this.value = 'Kabupaten/Kota Fasilitas Kesehatan Peserta Terdaftar';" />
    </div>
    <div class="input-group">
      <input type="text" id="pnk10" name="pnk10" required="required" placeholder="Tipe faskes *"
        onFocus="if (this.value == 'Tipe faskes ') this.value = '';"
        onBlur="if (this.value == '') this.value = 'Tipe faskes ';" />
    </div>
    <div class="input-group">
      <input type="text" id="pnk15" name="pnk15" required="required" placeholder="Nama Diagnosis *"
        onFocus="if (this.value == 'Nama Diagnosis') this.value = '';"
        onBlur="if (this.value == '') this.value = 'Nama Diagnosis';" />
    </div>
    <div class="comment_note">All fields marked with an asterisk (*) are required</div>
    <button type="submit" class="btn">predict</button>
    <div class="col-md-7">
      <hr>
      <div class="result">
        <h3><b>Result:</b> {{finalData}}</h3>
      </div>
    </div>
  </form>
</section>
<!-- system Section end -->

```

Gambar index html

Gambar tampilan index

6.2 Mempersiapkan Script dan Code

Sebelum tahapan code, perlu dilakukan penginstalan flask beberapa library yang dibutuhkan, lalu menggunakan virtual environment yang dimana merupakan lokasi library yang dibutuhkan tersebut dikelola.

6.2.1 Install Flask

Pada proyek ini dilakukan install flask yang bertujuan untuk membantu dalam menyediakan. Flask adalah sebuah web framework yang ditulis dengan bahasa Python dan tergolong sebagai jenis microframework. Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu web. Dengan menggunakan Flask dan bahasa Python, pengembang dapat membuat sebuah web yang terstruktur dan dapat mengatur behaviour suatu web dengan lebih mudah. Flask dapat diinstall dengan perintah berikut.

```
pip install flask
```

6.2.2 Create App.py

File app.py akan digunakan sebagai controller untuk menjalankan sistem, berikut merupakan code dari app.py

```
1  from flask import Flask, render_template, request, redirect
2  import pickle
3  import sklearn
4  import numpy as np
5
6  app = Flask(__name__)
7
8
9  @app.route('/')
10 def home():
11     return render_template('index.html', insurance_cost=2)
12
13
14 @app.route('/predict', methods=['POST','GET'])
15 def index():
16     if request.method == 'POST':
17
18         with open('model.pkl','rb') as r:
19             model = pickle.load(r)
20
21         pstv01 = float(request.form['pstv01'])
22         pstv10 = float(request.form['pstv10'])
23         pstv14 = float(request.form['pstv14'])
24         pnk10 = float(request.form['pnk10'])
25         pnk15 = float(request.form['pnk15'])
26
27         datas = np.array((pstv01,pstv10,pstv14,pnk10,pnk15))
28         datas = np.reshape(datas,(1,-1))
29
30         isLayanan = model.predict(datas)
31
32         if isLayanan == 1:
33             output = "RITP"
34         elif isLayanan == 0:
35             output = "RJTP"
36         else:
37             output = 'PROMOTIF'
38
39         return render_template('index.html', finalData=output)
40 if __name__ == '__main__':
41     app.run(debug=True)
```

Pada app.py terdapat potongan code sebagai berikut yang digunakan untuk mengimport library

```
from flask import Flask, render_template, request, redirect
import pickle
import sklearn
import numpy as np
```

Berikut merupakan potongan code full pada method POST

```
@app.route('/predict', methods=['POST','GET'])
def index():
    if request.method == 'POST':

        with open('model.pkl','rb') as r:
            model = pickle.load(r)

        pstv01 = float(request.form['pstv01'])
        pstv10 = float(request.form['pstv10'])
        pstv14 = float(request.form['pstv14'])
        pnk10 = float(request.form['pnk10'])
        pnk15 = float(request.form['pnk15'])

        datas = np.array((pstv01,pstv10,pstv14,pnk10,pnk15))
        datas = np.reshape(datas,(1,-1))

        isLayanan = model.predict(datas)

        if isLayanan == 1:
            output = "RITP"
        elif isLayanan == 0:
            output = "RJTP"
        else:
            output = 'PROMOTIF'

        return render_template('index.html', finalData=output)
if __name__ == '__main__':
    app.run(debug=True)
```

@app.route dengan method GET akan mengarahkan kita ke halaman index.html yang berisi form untuk dapat mengisi nilai-nilai setiap atribut yang dibutuhkan untuk melakukan klasifikasi.

```
@app.route('/')
def home():
    return render_template('index.html', insurance_cost=2)
```

Apabila user telah mengisi form dan memiliki button predict, maka user akan diarahkan ke route (/predict) dengan method POST.

```
@app.route('/predict', methods=['POST','GET'])
def index():
    if request.method == 'POST':
```

Dengan mendefinisikan index, jika request post diterima dari user yang mengisi form, maka model.pkl yang telah disimpan pada folder yang sama akan dibuka, dan akan menerima request atau inputan nilai yang diterima dari form yaitu pstv01, pstv10, pstv14, pnk10 dan pnk15 variabel di atas akan menampung kelima inputan tersebut, dan variabel prediksi akan melakukan prediksi

menggunakan model terhadap variabel datas yang dimana tadi berisi inputan yang diberikan oleh user melalui form

```
@app.route('/predict', methods=['POST','GET'])
def index():
    if request.method == 'POST':

        with open('model.pkl','rb') as r:
            model = pickle.load(r)

        pstv01 = float(request.form['pstv01'])
        pstv10 = float(request.form['pstv10'])
        pstv14 = float(request.form['pstv14'])
        pnk10 = float(request.form['pnk10'])
        pnk15 = float(request.form['pnk15'])

        datas = np.array((pstv01,pstv10,pstv14,pnk10,pnk15))
        datas = np.reshape(datas,(1,-1))

        isLayanan = model.predict(datas)
```

setelah itu jika prediksi ==1 maka outputnya adalah RITP, jika 0 maka outputnya RJTP, dan jika selain 0 dan 1 maka outputnya adalah PROMOTIF.

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Data Mining merupakan proses penambangan atau penggalian informasi dari data yang besar. Dalam melakukan proses data mining, perlu mempersiapkan data yang terlebih dahulu dengan melalui beberapa tahapan data preprocessing yaitu data integration, data selection, memetakan nomor kolom menjadi nama kolom, data cleaning, dan data transformation. Setelah data dipersiapkan dan dibersihkan, selanjutnya membangun model klasifikasi menggunakan Artificial Neural Network. Langkah terakhir yang dilakukan adalah deployment, tim mengembangkan sistem sederhana dengan menggunakan python flask.

7.2 Saran

Optimalkan akurasi model Artificial Neural Network (ANN) untuk meningkatkan kemampuannya dalam memprediksi tingkat pelayanan fasilitas yang diterima oleh peserta Fasilitas Kesehatan Tingkat Pertama pengguna BPJS yang mengidap Tuberkolosis.