

DOKUMEN PROYEK DATA MINING

12S4054 – PENAMBANGAN DATA

***Classification Kepesertaan pada Data BPJS Kesehatan
Tahun 2015-2021 using Random Forest***



Disusun oleh :

12S20019 Kristina Margaret Sitorus

12S20042 Matawila Febryanti Simanjuntak

12S20043 Sevia Rajagukguk

12S20052 Eka Rohani Gultom

PROGRAM STUDI SARJANA SISTEM INFORMASI

FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO

INSTITUT TEKNOLOGI DEL

2023

DAFTAR ISI

BAB I	5
1.1 Determine Business Objective	5
1.2 Determine Project Goal	6
1.3 Produce Project Plan	6
BAB 2.....	8
2.1 Collecting Data	8
2.2 Describe Data.....	8
2.3 Validation Data	10
BAB 3.....	12
3.1 Data Selection	12
3.2 Data Cleaning	13
3.3 Data Construct	14
3.4 Labeling Data.....	16
3.5 Data Integration.....	18
BAB 4.....	20
4.1 Building Test Scenario	20
4.2 Random Forest	21
4.3 Build Model.....	21
BAB 5.....	24
5.1 Evaluate Result	24
5.2 Evaluate Process.....	25
BAB 6.....	26
6.1 Membuat Rencana Deployment Model.....	26
6.2 Melakukan Deployment Model	26

DAFTAR TABEL

Table 1. Jadwal Pelaksanaan Proyek	6
Table 2. Table Spesifikasi.....	7
Table 3. Tabel Atribut	8

DAFTAR GAMBAR

Gambar 1. Validation Data	11
Gambar 2. Data Selection	12
Gambar 3. Output Data Selection	12
Gambar 4. Data Cleaning	13
Gambar 5. Ouput Data Cleansing	14
Gambar 6. Data Construct	15
Gambar 7. Output Data Construct.....	16
Gambar 8. Labelling Data.....	16
Gambar 9. Ouput Labelling Data.....	17
Gambar 10. Data Integration	19
Gambar 11. Ouput Data Integration	19
Gambar 12. Build Model	22
Gambar 13. Import Dataset.....	22
Gambar 14. Mendefenisikan x dan y	22
Gambar 15. Membagi data train dan data test.....	23
Gambar 16. Melakukan Train.....	23
Gambar 17. Confussion Matrix	24
Gambar 18. Kode Perintah app.py	28
Gambar 19. Tampilan Hasil Deploymen.....	28

BAB I

BUSINESS UNDERSTANDING

Langkah pertama dalam melakukan klasifikasi kepesertaan anggota BPJS adalah dengan melakukan *business understanding*. Sehingga pada bab ini akan dijelaskan terkait aktivitas pada data mining untuk meningkatkan pemahaman diantaranya adalah menentukan objektif bisnis, menentukan tujuan bisnis dan membuat rencana proyek.

1.1 Determine Business Objective

Perkembangan teknologi yang sangat pesat telah membantu masyarakat mendapatkan informasi dengan mudah. Informasi tersebut diantaranya adalah informasi kesehatan. Kesehatan merupakan hal yang penting bagi setiap orang dan pemerintah bertanggung jawab untuk memastikan bahwa setiap masyarakat memperoleh pelayanan kesehatan dengan baik dimanapun dan kapanpun. Maka dari itu diperlukan pendataan anggota BPJS agar dapat dilihat persebaran fasilitas kesehatan yang harus dipersiapkan di setiap daerah. Pendataan tersebut dilakukan agar mengetahui status peserta BPJS baik anggota aktif maupun tidak aktif. Pendataan tersebut dapat dilakukan melalui *classification* dengan algoritma *random forest*. *Classification* merupakan pengelompokan sebuah data ke dalam suatu kategori tertentu dengan melihat karakteristik tertentu. *Classification* dengan menggunakan algoritma *random forest* pada *machine learning* yang memanfaatkan teknik *ensemble learning*. *Ensemble learning* melibatkan penggabungan beberapa model pada *machine learning* yang digunakan untuk tugas klasifikasi maupun regresi. Dengan beberapa kelebihan algoritma *random forest*, algoritma tersebut juga memiliki kelemahan dimana cenderung menyebabkan data *overfitting* atau *noise*, membutuhkan banyak memori dan waktu, imbalanced data dan tidak tepat dalam menangani masalah dengan dimensi waktu.

Data kepesertaan merupakan sebuah kumpulan data kategorial yang berisi data peserta BPJS. Data tersebut perlu dianalisis sebelum dijadikan model yang akan dibangun dan diimplementasikan menggunakan metode atau algoritma *random forest*. Sehingga dapat dijelaskan objektif yang akan dicapai dalam pengerjaan proyek ini adalah

1. Memeriksa apakah terdapat *missing* atau *duplicate* data. Data yang sesuai akan dilakukan analisis *Exploratory Data Analysis* (EDA). Data yang besar akan memungkinkan terjadinya redundancy data.

2. Meningkatkan performa model menggunakan data kepesertaan BPJS. Meningkatkan performa model dibanding dengan performa model sebelumnya.

1.2 Determine Project Goal

Tujuan pengerjaan proyek ini adalah membangun sebuah model dengan menggunakan teknik pada data mining untuk mengetahui klasifikasi peserta BPJS pada tahun 2015 - 2021 dengan luaran adalah informasi apakah peserta tersebut merupakan peserta BPJS aktif maupun sudah tidak aktif.

1.3 Produce Project Plan

Tahap perencanaan yang dilakukan untuk mencapai tujuan pengerjaan proyek penelitian ini dengan membuat susunan dan rencana proyek secara terstruktur. Dengan timeline yang tepat dengan waktu yang telah ditentukan. Berikut ini adalah susunan rencana pengerjaan proyek sebagai berikut

Table 1. Jadwal Pelaksanaan Proyek

Tahapan	Waktu Pengerjaan	Kegiatan
<i>Business Understanding</i>	3 hari	Menentukan objektif bisnis, menentukan tujuan proyek serta membuat rencana proyek.
<i>Data Understanding</i>	3 hari	Mengumpulkan data yang akan digunakan, menelaah data dan melakukan validasi pada data.
<i>Data Preparation</i>	4 hari	Memilih data yang akan digunakan, membersihkan data, mengkonstruksi data, menentukan label data, dan mengintegrasikan data.
<i>Modeling</i>	3 hari	Membangun skenario pengujian dan membangun model.

Tahapan	Waktu Pengerjaan	Kegiatan
<i>Evaluation</i>	3 hari	Melakukan evaluasi hasil pemodelan dan melakukan review terhadap proses pemodelan.
<i>Deployment</i>	4 hari	Membuat rencana deployment model, Monitoring and Maintenance rencana deployment model dan meninjau proyek.

Selama proses pengerjaan proyek penelitian ini diperlukan spesifikasi diantaranya adalah sebagai berikut

Table 2. Table Spesifikasi

<i>Tools</i>	<ul style="list-style-type: none"> - <i>Jupyter Notebook</i> - <i>Google Collab</i> - <i>Visual Studio Code</i>
Bahasa Pemrograman	<i>Python</i>
Algoritma	<i>Random Forest</i>
<i>Web Interface</i>	<i>Flask Python</i>
<i>Deployment Cloud</i>	<i>Visual Studio Code</i>

BAB 2

DATA UNDERSTANDING

Dalam tahapan data understanding yang merupakan tahapan pemahaman terhadap data yang akan digunakan, tahapan ini dimulai dari mengumpulkan data, mendeskripsikan data dan memahami data yang akan digunakan dalam penelitian.

2.1 Collecting Data

Pengumpulan data merupakan tahap awal untuk menemukan data yang akan digunakan dalam penelitian. maka dari itu dataset yang akan digunakan untuk klasifikasi status kepesertaan pada data kepesertaan bpjs kesehatan mulai tahun 2015-2021 berdasarkan dataset yaitu data sampel BPJS Kesehatan tahun 2015-2021 sebesar 2.305.435 peserta. Beberapa data yang digunakan pada tugas ini adalah file kepesertaan Diabetes Melitus pada tahun 2019, 2020, 2021 dan file kepesertaan Tuberculosis tahun 2019, 2020, 2021 serta file reguler kepesertaan.

2.2 Describe Data

Menelaah data pada data understanding adalah proses untuk mendapatkan pemahaman awal mengenai data dengan melakukan analisis dan visualisasi data. Data sampel kepesertaan kontekstual Diabetes Mellitus (DM) adalah seluruh peserta tersampel yang ada dalam sistem informasi BPJS Kesehatan sebagai representative peserta JKN-KIS pernah didiagnosis DM. Data kepesertaan terkait karakteristik selain jenis kelamin bersifat dinamis, dan data kepesertaan berikut menunjukkan status peserta pada tanggal 31 Desember 2021. Hasil analisis pada tabel adalah jumlah dan persentase tertimbang terhadap populasi menggunakan variabel bobot. Berikut tabel yang membahas terkait atribut pada dataset kepesertaan pada file DM 2019 kepesertaan.dta

Table 3. Tabel Atribut

No.	Variable	Nama variable	Tipe variable	
1.	PSTV01	Nomor peserta	Integer	Kategorik (Nominal)

No.	Variable	Nama variable	Tipe variable	
2.	PSTV02	Nomor keluarga	Integer	Kategorik (Nominal)
3.	PSTV03	Tanggal lahir peserta	String	Numerik (Interval-scaled)
4.	PSTV04	Hubungan keluarga	String	Kategorik (Nominal)
5.	PSTV05	Jenis kelamin	String	Kategorik (Nominal)
6.	PSTV06	Status perkawinan	String	Kategorik (Nominal)
7.	PSTV07	Kelas rawat	String	Kategorik (Nominal)
8.	PSTV08	Segmentasi peserta	String	Kategorik (Nominal)
9.	PSTV09	Provinsi tempat tinggal peserta	String	Kategorik (Nominal)
10.	PSTV10	Kabupaten/Kota Tempat Tinggal Peserta	String	Kategorik (Nominal)

No.	Variable	Nama variable	Tipe variable	
11.	PSTV11	Kepemilikan faskes	String	Kategorik (Nominal)
12.	PSTV12	Jenis faskes	String	Kategorik (Nominal)
13.	PSTV13	"Provinsi Fasilitas Kesehatan Peserta Terdaftar	String	Kategorik (Nominal)
14.	PSTV14	Kabupaten/Kota Fasilitas Kesehatan Peserta Terdaftar	String	Kategorik (Nominal)
15.	PSTV15	Bobot	Float	Numerik (Ratio)
16.	PSTV16	Tahun Sample	Integer	Numerik (Ratio)
17.	PSTV17	Status Kepesertaan	String	Numerik (Ordinal)
18.	PSTV18	Tahun Meninggal	Integer	Numerik (Ratio)

2.3 Validation Data

Pada tahap ini dilakukan validasi terhadap data yang akan digunakan data yang akan digunakan dengan memeriksa kelengkapan data untuk menghindari terjadinya *error* ataupun masalah *input data* yang terjadi *missing value*. Maka dari informasi ini, kita dapat melihat nama kolom, tipe data, jumlah non-null entries, dan penggunaan memori DataFrame. Hal ini berguna untuk memahami struktur dan ukuran data, serta mengevaluasi apakah ada nilai yang hilang atau tipe data yang perlu diubah sesuai kebutuhan analisis data atau proses machine learning.

Validation Data

```
print(merge_df.shape)
merge_df.info()
```

(3020821, 18)
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3020821 entries, 0 to 2305434
Data columns (total 18 columns):
Column Dtype
--- ---
0 PSTV01 int32
1 PSTV02 int32
2 PSTV03 datetime64[ns]
3 PSTV04 category
4 PSTV05 category
5 PSTV06 object
6 PSTV07 object
7 PSTV08 object
8 PSTV09 category
9 PSTV10 object
10 PSTV11 object
11 PSTV12 object
12 PSTV13 object
13 PSTV14 object
14 PSTV15 float32
15 PSTV16 int16
16 PSTV17 object
17 PSTV18 float64
dtypes: category(3), datetime64[ns](1), float32(1), float64(1), int16(1), int32(2), object(9)
memory usage: 325.5+ MB

Gambar 1. Validation Data

BAB 3

DATA PREPARATION

Data Preparation akan dilakukan untuk menghasilkan data yang memiliki kualitas baik. Berdasarkan penjelasan dari Bab 2 , data preparation dilakukan dengan beberapa tahapan meliputi Data Selection, *Data Cleaning*, *Data Transformation*, *Data Selection (Feature Selection)*, *Data Labelling* dan *Data Integration*.

3.1 Data Selection

Dalam proses memilih data adalah memegang peranan penting dalam penyelesaian kasus data BPJS Kesehatan tahun 2015-2021 demi keperluan dalam proses pengklasifikasian menggunakan *Random Forest*. Memilih data menjadi bagian pelatihan dan pengujian dengan memberikan dasar untuk melatih dan menguji model secara terpisah. Pentingnya pemilahan data terlihat saat menguji model pada sebagian kecil data yang tidak digunakan selama pelatihan. Ini memberikan pemahaman yang lebih objektif tentang sejauh mana model dapat memprediksi kepemilikan. Keseluruhan, pemilahan data bukan hanya langkah persiapan, melainkan fondasi utama untuk membangun model klasifikasi yang handal dan akurat pada *Classification* Kepesertaan dalam Data BPJS Kesehatan Tahun 2015-2021 menggunakan Random Forest.

```
to_drop = ['PSTV01', 'PSTV03']
merge_encode.drop(to_drop, inplace=True, axis = 1)
merge_encode
```

Gambar 2. Data Selection

	PSTV02	PSTV04	PSTV05	PSTV06	PSTV07	PSTV08	PSTV09	PSTV10	PSTV11	PSTV12	PSTV13	PSTV14	PSTV15	PSTV16	PSTV17
0	244485	2	1	2	1	6	0	402	2	3	0	402	72364	2020	AKTIF
1	985323	2	0	2	1	6	0	171	2	3	0	171	58346	2020	AKTIF
2	1009059	1	1	1	1	1	33	42	2	3	33	42	48412	2020	AKTIF
3	478252	2	1	2	1	6	33	13	2	3	33	13	65483	2020	MENINGGAL
4	928532	3	0	2	1	1	31	401	2	3	31	401	63809	2020	AKTIF
...
2305430	904168	2	0	2	3	4	8	260	2	3	8	260	211530	2019	AKTIF
2305431	1028209	0	1	0	2	6	10	127	5	0	10	428	117447	2019	AKTIF
2305432	1004884	2	1	2	3	4	8	18	2	3	8	18	119617	2019	AKTIF
2305433	967284	2	1	2	3	4	9	458	2	3	9	458	182818	2019	AKTIF
2305434	985262	2	0	2	3	4	9	396	2	3	9	396	142396	2019	TIDAK AKTIF

2907704 rows x 15 columns

Gambar 3. Output Data Selection

Alasan dilakukan drop PSTV01 dikarenakan index dari pendataan kepesertaan, sementara PSTV03 merupakan tanggal lahir dari peserta, sehingga kurang relevan apabila digunakan sebagai inputan pada model

3.2 Data Cleaning

Data cleaning adalah proses mengidentifikasi dan memperbaiki (atau menghapus) kesalahan, inkonsistensi, dan ketidaksesuaian dalam set data. Tujuan utama dari data cleaning adalah untuk meningkatkan kualitas data, sehingga data tersebut dapat diandalkan dan digunakan untuk analisis atau pengambilan keputusan yang akurat. Proses ini penting karena data yang kotor atau tidak sesuai dapat menghasilkan hasil yang tidak akurat. Kita dapat membersihkan data dengan menghapus missing value yang ada pada data. Selain itu dapat di cek duplicate value yang ada pada data.

```
# checking missing value
merge_df.isna().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    2909754
dtype: int64
```

Gambar 4. Data Cleaning

Gambar diatas merupakan hasil (output) dari pengecekan *missing value* yang terdapat pada data. Berdasarkan output missing value di atas, kolom PSTV18 memiliki nilai yang cenderung kosong. Maka akan dilakukan drop terhadap kolom sebagai solusinya.

```
merge_df.drop(['PSTV18'], inplace=True, axis = 1)
merge_df
```

	PSTV01	PSTV02	PSTV03	PSTV04	PSTV05	PSTV06	PSTV07	PSTV08	PSTV09	PSTV10	PSTV11	PSTV12	PSTV13	PSTV14	PSTV15	PSTV16	PSTV17
0	45243428	45243428	1959-10-11	PESERTA	PEREMPUAN	KAWIN	KELAS I	PPU	ACEH	PIDIE	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	PIDIE	11.468968	2020	AKTIF
1	356470819	356470819	1965-12-31	PESERTA	LAKI-LAKI	KAWIN	KELAS I	PPU	ACEH	KOTA BANDA ACEH	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	KOTA BANDA ACEH	9.863322	2020	AKTIF
2	72280409	375793382	1964-08-03	ISTRI	PEREMPUAN	CERAI	KELAS I	BUKAN PEKERJA	SUMATERA UTARA	BATU BARA	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	SUMATERA UTARA	BATU BARA	8.487743	2020	AKTIF
3	88501975	88501975	1959-10-02	PESERTA	PEREMPUAN	KAWIN	KELAS I	PPU	SUMATERA UTARA	ASAHAN	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	SUMATERA UTARA	ASAHAN	10.726228	2020	MENINGGAL
4	94870095	316527655	1947-01-01	SUAMI	LAKI-LAKI	KAWIN	KELAS I	BUKAN PEKERJA	SUMATERA BARAT	PESISIR SELATAN	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	SUMATERA BARAT	PESISIR SELATAN	10.539836	2020	AKTIF
...
2305430	290698984	290698984	1965-05-14	PESERTA	LAKI-LAKI	KAWIN	KELAS III	PBI APBN	JAWA BARAT	KOTA TASIKMALAYA	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	JAWA BARAT	KOTA TASIKMALAYA	3312.561035	2019	AKTIF
2305431	402778434	391518127	2013-03-17	ANAK	PEREMPUAN	BELUM KAWIN	KELAS II	PPU	JAWA TIMUR	JEMBER	SWASTA	DOKTER UMUM	JAWA TIMUR	SAMPANG	54.075462	2019	AKTIF
2305432	372419310	372419310	1958-02-10	PESERTA	PEREMPUAN	KAWIN	KELAS III	PBI APBN	JAWA BARAT	BANDUNG BARAT	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	JAWA BARAT	BANDUNG BARAT	60.153950	2019	AKTIF
2305433	341838920	341838920	1958-03-26	PESERTA	PEREMPUAN	KAWIN	KELAS III	PBI APBN	JAWA TENGAH	SRAGEN	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	JAWA TENGAH	SRAGEN	465.500214	2019	AKTIF
2305434	356423412	356423412	1943-01-10	PESERTA	LAKI-LAKI	KAWIN	KELAS III	PBI APBN	JAWA TENGAH	PEMALANG	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	JAWA TENGAH	PEMALANG	137.629379	2019	TIDAK AKTIF

3020821 rows x 17 columns

Gambar 5. Ouput Data Cleansing

```
merge_df.dropna()
```

Selanjutnya dilakukan pengecekan data yang duplikat.

```
# checking duplicate value
merge_df.duplicated().sum()

113117
```

Kode tersebut digunakan untuk menghitung jumlah baris yang merupakan duplikat dalam DataFrame merge_df. Jika hasilnya adalah 0, itu berarti tidak ada baris yang duplikat. Jika hasilnya lebih dari 0, itu menunjukkan bahwa ada baris-baris duplikat dalam DataFrame tersebut. Pengecekan duplikat ini dapat membantu dalam membersihkan dan memastikan kebersihan data sebelum melakukan analisis lebih lanjut.

```
merge_df.drop_duplicates(inplace=True)
```

```
print("Jumlah duplikasi: ", merge_df.duplicated().sum())
```

Kode ini berfungsi untuk menghapus baris-baris yang merupakan duplikat dari DataFrame merge_df dan mengubahnya secara langsung. Hasilnya adalah DataFrame yang telah dibersihkan dari duplikat, dan perubahan tersebut diterapkan pada variabel merge_df.

3.3 Data Construct

Data construction adalah salah satu tahapan dalam persiapan data (data preparation) yang melibatkan pembuatan atau transformasi variabel-variabel baru berdasarkan data yang sudah ada. Dalam proses ini, tujuan utamanya adalah untuk meningkatkan relevansi atau

informativitas data yang akan digunakan dalam analisis lebih lanjut. Data construction dapat mencakup beberapa kegiatan, seperti penggabungan variabel, pembuatan variabel turunan, atau pengelompokan data. Data construction juga dapat membantu mengatasi masalah keterbatasan informasi yang ada dengan menciptakan variabel yang lebih informatif atau relevan untuk tujuan analisis yang spesifik.

```
def label_encode_columns(dataframe, columns):
    le = LabelEncoder()
    for column in columns:
        # Check if the column contains any non-numeric values
        if not pd.api.types.is_numeric_dtype(dataframe[column]):
            # If not numeric, convert the column to string before applying LabelEncoder
            dataframe[column] = dataframe[column].astype(str)

        dataframe[column] = le.fit_transform(dataframe[column])
    return dataframe

# Example usage
columns_to_encode = ['PSTV02', 'PSTV04', 'PSTV05', 'PSTV06', 'PSTV07', 'PSTV08', 'PSTV09', 'PSTV10', 'PSTV11',
                    'PSTV12', 'PSTV13', 'PSTV14', 'PSTV15']

merge_encode = label_encode_columns(merge_df, columns_to_encode)
```

Gambar 6. Data Construct

Kode tersebut merupakan implementasi dari suatu fungsi yang disebut `label_encode_columns`. Fungsi ini bertujuan untuk melakukan label encoding pada kolom-kolom tertentu dalam suatu DataFrame. Label encoding adalah proses menggantikan nilai-nilai kategorikal atau teks dengan nilai-nilai numerik. Pertama, dalam fungsi ini, objek `LabelEncoder` dari modul scikit-learn diinisialisasi. Selanjutnya, dilakukan iterasi pada setiap kolom yang terdaftar dalam parameter `columns`. Untuk setiap kolom, dilakukan pengecekan apakah tipe datanya sudah numerik atau belum. Jika belum, kolom tersebut diubah menjadi tipe data string agar dapat di-label encode. Kemudian, label encoder diterapkan pada kolom tersebut, menggantikan nilai-nilai kategorikal dengan nilai numerik yang sesuai. Fungsi mengembalikan DataFrame yang telah diubah. Dalam contoh penggunaan di bagian akhir, kita menyertakan DataFrame `merge_df` dan daftar kolom `columns_to_encode` yang ingin di-label encode. Hasilnya disimpan dalam variabel `merge_encode`. Dengan melakukan ini, kita dapat mengubah representasi nilai kategorikal menjadi format numerik, yang seringkali diperlukan dalam analisis data atau pemodelan.

```
merge_encode.head()
```

	PSTV01	PSTV02	PSTV03	PSTV04	PSTV05	PSTV06	PSTV07	PSTV08	PSTV09	PSTV10	PSTV11	PSTV12	PSTV13	PSTV14	PSTV15	PSTV16
0	45243428	244485	1959-10-11	2	1	2	1	6	0	402	2	3	0	402	72364	2020
1	356470819	985323	1965-12-31	2	0	2	1	6	0	171	2	3	0	171	58346	2020
2	72280409	1009059	1964-08-03	1	1	1	1	1	33	42	2	3	33	42	48412	2020
3	88501975	478252	1959-10-02	2	1	2	1	6	33	13	2	3	33	13	65483	2020
4	94870095	928532	1947-01-01	3	0	2	1	1	31	401	2	3	31	401	63809	2020

Gambar 7. Output Data Construct

3.4 Labeling Data

Labelling data adalah proses memberikan atau menetapkan label atau kategori tertentu pada setiap instance atau entitas dalam kumpulan data. Tujuan utama dari labelling data adalah untuk memberikan identifikasi atau makna pada data, sehingga memungkinkan algoritma pembelajaran mesin atau model statistik untuk belajar dan membuat prediksi dengan lebih baik. Misalnya, dalam masalah klasifikasi, setiap instance dalam data diberi label kelas tertentu yang mencerminkan kategori atau kelompok yang ingin diprediksi oleh model. Labelling data sangat penting untuk melatih model dan mengevaluasi kinerjanya, karena model dapat memahami pola atau relasi antara fitur-fitur dan label yang sudah diberikan.

```
merge_encode['PSTV17'].value_counts()
```

```

AKTIF          1948858
TIDAK AKTIF    856018
MENINGGAL      102735
99              92
30               1
Name: PSTV17, dtype: int64

```

Gambar 8. Labelling Data

Kode tersebut adalah perintah Python yang menggunakan metode `value_counts()` pada kolom 'PSTV17' dari objek DataFrame yang disebut `merge_encode`. Output yang diberikan menunjukkan distribusi nilai unik dalam kolom 'PSTV17' beserta jumlah kemunculannya. Berikut adalah penjelasan untuk output yang diberikan:

- AKTIF: Terdapat 1.948.858 baris dengan nilai 'AKTIF' dalam kolom 'PSTV17'.
- TIDAK AKTIF: Terdapat 856.018 baris dengan nilai 'TIDAK AKTIF' dalam kolom 'PSTV17'.

- MENINGGAL:** Terdapat 102.735 baris dengan nilai 'MENINGGAL' dalam kolom 'PSTV17'.
- Terdapat 92 baris dengan nilai '99' dalam kolom 'PSTV17'.
- Terdapat 1 baris dengan nilai '30' dalam kolom 'PSTV17'.

Dengan kata lain, output tersebut memberikan informasi tentang seberapa sering setiap nilai khusus muncul dalam kolom 'PSTV17'. Hal ini dapat membantu dalam memahami distribusi data dan melihat proporsi masing-masing kategori dalam kolom tersebut.

```
[ ] merge_encode['PSTV17'] = merge_encode['PSTV17'].map({'AKTIF': 1, 'TIDAK AKTIF': 0, 'MENINGGAL': 0, '99': 0, '30': 0})

[ ] merge_encode['PSTV17'].value_counts()

1.0    1948858
0.0     958753
Name: PSTV17, dtype: int64
```

```
merge_encode.head()
```

	PSTV02	PSTV04	PSTV05	PSTV06	PSTV07	PSTV08	PSTV09	PSTV10	PSTV11	PSTV12	PSTV13	PSTV14	PSTV15	PSTV16	PSTV17
0	244485	2	1	2	1	6	0	402	2	3	0	402	72364	2020	1.0
1	985323	2	0	2	1	6	0	171	2	3	0	171	58346	2020	1.0
2	1009059	1	1	1	1	1	33	42	2	3	33	42	48412	2020	1.0
3	478252	2	1	2	1	6	33	13	2	3	33	13	65483	2020	0.0
4	928532	3	0	2	1	1	31	401	2	3	31	401	63809	2020	1.0

Gambar 9. Ouput Labelling Data

Kode ini menggunakan metode `value_counts()` pada kolom 'PSTV17' dari objek DataFrame yang disebut `merge_encode`. Output yang dihasilkan menunjukkan distribusi nilai unik dalam kolom 'PSTV17' beserta jumlah kemunculannya. Berikut adalah penjelasan untuk output yang diberikan:

- 1.0: Terdapat 1.948.858 baris dengan nilai 1.0 dalam kolom 'PSTV17'.
- 0.0: Terdapat 958.753 baris dengan nilai 0.0 dalam kolom 'PSTV17'.

Dengan kata lain, output tersebut memberikan informasi tentang seberapa sering setiap nilai khusus (1.0 dan 0.0) muncul dalam kolom 'PSTV17'. Ini dapat memberikan wawasan tentang distribusi proporsi antara dua nilai tersebut dalam kolom tersebut. Misalnya, dalam konteks biner seperti ini, mungkin 1.0 dan 0.0 merujuk pada dua kondisi atau kategori tertentu, dan output tersebut mencerminkan berapa banyak data yang termasuk dalam masing-masing kategori tersebut.

```
merge_encode.isna().sum()
```

```
PSTV02    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    93
dtype: int64
```

```
clean_df = merge_encode.dropna()
clean_df.head()
```

	PSTV02	PSTV04	PSTV05	PSTV06	PSTV07	PSTV08	PSTV09	PSTV10	PSTV11	PSTV12	PSTV13	PSTV14	PSTV15	PSTV16	PSTV17
0	244485	2	1	2	1	6	0	402	2	3	0	402	72364	2020	1.0
1	985323	2	0	2	1	6	0	171	2	3	0	171	58346	2020	1.0
2	1009059	1	1	1	1	1	33	42	2	3	33	42	48412	2020	1.0
3	478252	2	1	2	1	6	33	13	2	3	33	13	65483	2020	0.0
4	928532	3	0	2	1	1	31	401	2	3	31	401	63809	2020	1.0

```
clean_df.isna().sum()
```

```
PSTV02    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
dtype: int64
```

3.5 Data Integration

Pada tahapan mengintegrasikan data dilakukan dengan tahapan awal adalah membentuk dataset yang lebih lengkap dan siap digunakan dalam pembangunan model klasifikasi. Pada data integration ini kita menggabungkan informasi dari berbagai sumber tahun ke dalam satu dataset yang lebih besar yang bertujuan dapat meningkatkan kemampuan model dalam memahami pola umum dalam data kepesertaan BPJS Kesehatan sepanjang tahun 2015-2021.

Dapat dilihat pada tahapan ini data dilakukan *merge concatenation*. Dimana ini merupakan proses pendekatan untuk membentuk satu dataset dan menyimpan DataFrames terpisah untuk setiap tahun atau kategori kepesertaan guna persiapan untuk proses concatenation yaitu dengan membentuk satu dataset yang lebih besar yang lebih baik yang mencakup data kepesertaan BPJS tahun 2015-2021 dan kategori.

```
[ ] df = [DM2019_kepesertaan,DM2020_kepesertaan,DM2021_kepesertaan,TB2019_kepesertaan,TB2020_kepesertaan]
merge_df = pd.concat(df)
```

Gambar 10. Data Integration

```
[ ] print(merge_df.shape)
merge_df.head()
```

Dari code `print(merge_df.shape)` bertujuan memastikan bahwa proses concatenation berjalan dengan benar dan mendapatkan informasi terkait ukuran dataset.

	PSTV01	PSTV02	PSTV03	PSTV04	PSTV05	PSTV06	PSTV07	PSTV08	PSTV09	PSTV10	PSTV11	PSTV12	PSTV13	PSTV14	PSTV15	PSTV16	PSTV17	PSTV18
0	45243428	45243428	1959-10-11	PESERTA	PEREMPUAN	KAWIN	KELAS I	PPU	ACEH	PIDIE	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	PIDIE	11.468968	2020	AKTIF	MAN
1	356470819	356470819	1965-12-31	PESERTA	LAKI-LAKI	KAWIN	KELAS I	PPU	ACEH	KOTA BANDA ACEH	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	ACEH	KOTA BANDA ACEH	9.863322	2020	AKTIF	MAN
2	72280409	375793382	1964-08-03	ISTRI	PEREMPUAN	CERAI	KELAS I	BUKAN PEKERJA	SUMATERA UTARA	BATU BARA	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	SUMATERA UTARA	BATU BARA	8.487743	2020	AKTIF	MAN
3	88501975	88501975	1959-10-02	PESERTA	PEREMPUAN	KAWIN	KELAS I	PPU	SUMATERA UTARA	ASAHAN	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	SUMATERA UTARA	ASAHAN	10.726228	2020	MENINGGAL	2019.0
4	94870095	310527655	1947-01-01	SUAMI	LAKI-LAKI	KAWIN	KELAS I	BUKAN PEKERJA	SUMATERA BARAT	PESISIR SELATAN	PEMERINTAH KABUPATEN/KOTA	PUSKESMAS	SUMATERA BARAT	PESISIR SELATAN	10.539836	2020	AKTIF	MAN

Gambar 11. Ouput Data Integration

Dan dibawah ini memberikan tampilan ringkasan statistik deskriptif dari data, seperti mean, standar deviasi, nilai minimum, kuartil, dan nilai maksimum.

```
[ ] merge_df.describe()
```

	PSTV01	PSTV02	PSTV15	PSTV16	PSTV18
count	3.020821e+06	3.020821e+06	3.020821e+06	3.020821e+06	111067.000000
mean	1.434884e+08	1.424649e+08	8.893610e+01	2.017568e+03	2019.016179
std	1.203137e+08	1.213148e+08	2.353842e+02	1.961516e+00	1.669029
min	1.500000e+01	1.500000e+01	1.568037e-01	2.016000e+03	2014.000000
25%	5.104401e+07	4.962626e+07	8.332311e+00	2.016000e+03	2019.000000
50%	1.033770e+08	9.925007e+07	2.283859e+01	2.016000e+03	2019.000000
75%	1.990410e+08	1.985788e+08	8.855906e+01	2.020000e+03	2020.000000
max	4.605590e+08	4.605590e+08	5.386168e+04	2.021000e+03	2021.000000

BAB 4

MODELING

Pada bab ini merupakan bagian dalam pembangunan model dan pengimplementasian algoritma *Random Forest*. Pada proyek ini akan dibangun model dalam mengklasifikasikan peserta BPJS tahun 2015 - 2021. Classification merupakan proses pengelompokan data ke dalam kategori atau kelas tertentu berdasarkan atribut atau fitur yang dimiliki. Tujuan pada pengklasifikasian adalah untuk mengidentifikasi pola atau hubungan antara data sehingga dapat ditempatkan ke dalam suatu kategori atau kelas tertentu. Dalam hal ini, pengklasifikasian pada kepesertaan BPJS akan mengelompokkan peserta pada kelas peserta aktif dan tidak aktif. Pengklasifikasian memiliki beberapa tahapan pemodelan dari awal hingga proses hingga tahap deployment.

4.1 Building Test Scenario

Building test scenario merupakan pembangunan scenario kerja atau tahapan yang dilakukan pada tahap awal hingga evaluasi.

1. Data Understanding
 - Collecting data
 - Describe data
 - Validation data
2. Data Preparation
 - Data selection
 - Data cleaning
 - Data construct
 - Labelling data
 - Data Integration
3. Modelling
 - Building test scenario
 - Build model
4. Model Evaluation
 - Evaluate result
 - Evaluate process

Klasifikasi model yang diharapkan dapat mencapai persyaratan nilai dari setiap matriks evaluasi diantaranya adalah *precision* > 0.60 , *accuracy* > 0.60 dan *recall* > 0.65.

4.2 Random Forest

Random forest adalah suatu algoritma machine learning yang digunakan untuk tugas klasifikasi, regresi dan pengelompokan data. Pada klasifikasi random forest digunakan untuk memprediksi kelas suatu data. Misalnya pada case ini dapat menggunakan random forest untuk melihat sebuah data kepesertaan yang aktif dan tidak aktif.

Untuk melakukan klasifikasi, random forest pertama akan membangun kumpulan pohon keputusan, setiap pohon keputusan akan dibuat dengan menggunakan sampel data yang berbeda. sampel data yang dimaksud disini akan dipilih secara acak dari data keseluruhan. Keunggulan dari random forest ini adalah kinerja yang baik, tahan terhadap overfitting, dan kemampuan untuk menangani berbagai jenis data. Namun kelemahan dari random forest ini adalah membutuhkan banyak data dimana random forest membutuhkan banyak data untuk menghasilkan hasil yang akurat, dan membutuhkan waktu lama yang dimana untuk data yang besar.

4.3 Build Model

Pada tahapan ini merupakan proses membangun model setelah data melalui proses *understanding* dengan tahapan *collecting data* dan *validation data*. Kemudian melalui proses *preparation* dengan tahapan *data selection*, *data cleaning*, *labeling data* dan *data integration* hingga membangun *scenario test*. Pada tahapan telah dilakukan drop atau menghapus beberapa atribut maupun *feature* yang tidak relevan dan tidak dibutuhkan diantaranya adalah PSTV01 dan PSTV03 Alasan dilakukan drop PSTV01 dikarenakan index dari pendataan kepesertaan, sementara PSTV03 merupakan tanggal lahir dari peserta, sehingga kurang relevan apabila digunakan sebagai inputan pada model.

1. *Libraries import* adalah mengimport *library* yang dibutuhkan diantaranya adalah pandas, seaborn dan numpy

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.preprocessing import LabelEncoder
# Scaling
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
# Modelling
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
# Evaluate
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, precision_score, recall_score, classification_report

```

Gambar 12. Build Model

- pandas (pd) digunakan untuk menganalisis data dan menyediakan struktur data tingkat tinggi (DataFrame) yang memungkinkan pengolahan data yang efisien.
- matplotlib.pyplot (plt) digunakan untuk membuat visualisasi grafik dalam python. Pyplot adalah antarmuka yang menyediakan fungsi membuat berbagai jenis plot dan grafik.
- seaborn (sns) digunakan untuk membangun grafik statistik dan membuat visualisasi data yang lebih menarik dan informatif
- numpy (np) digunakan untuk menyediakan objek array dengan fungsi matematika.

2. Import Dataset

```

DM2019_kepesertaan=pd.read_stata ('/content/drive/MyDrive/DATA SET/DM2019_kepesertaan.dta')
DM2019_kepesertaan.head()

```

Gambar 13. Import Dataset

Pada baris pertama menggunakan fungsi *read_stata* untuk membaca file data dalam format stata dengan nama file DM2019_kepesertaan.dta. Data akan dibaca dan disimpan dalam sebuah variabel yang dinamakan DM2019_kepesertaan.

Pada baris kedua menggunakan fungsi *head()* untuk menampilkan beberapa baris awal dari suatu DataFrame pada pandas. Fungsi ini menampilkan gambaran , struktur dan konten pada data tersebut.

3. Mendefinisikan x dan y

```

X = clean_df.drop('PSTV17', axis = 1)
y = clean_df['PSTV17']

```

Gambar 14. Mendefinisikan x dan y

Mendefinisikan x pada PSTV17 dan y pada PSTV17

4. Membagi data train dan data test

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
X_train.shape, X_test.shape

((2326088, 14), (581523, 14))
```

Gambar 15. Membagi data train dan data test

Fungsi tersebut akan membagi dataset menjadi *data train* dan *data test*. Dimana 80% *training dataset* dan 20% *testing dataset* menggunakan *library* sklearn.

5. Melakukan Train

```
rfc=RandomForestClassifier()
rfc.fit(X_train,y_train)
y_pred=rfc.predict(X_test)
```

Gambar 16. Melakukan Train

Fungsi tersebut merupakan train dengan algoritma random forest dengan menggunakan library scikit-learn pada Python.

Pada baris pertama digunakan untuk membuat objek classifier menggunakan algoritma random forest dengan nama rfc (Random Forest Classifier) untuk menginisialisasi classifier dengan parameter default.

Pada baris kedua mengajarkan model memahami pola dalam data training. Dengan fungsi fit() untuk melatih model menggunakan dua parameter. x_train merupakan matriks fitur dari data latih dan y_train adalah vektor target yang sesuai dengan data latih.

Pada baris ketiga memprediksi nilai target untuk data testing menggunakan model yang telah dilatih sebelumnya. Dengan fungsi predict() menghasilkan prediksi berdasarkan matriks fitur x_test yang akan disimpan dalam variabel y_pred.

BAB 5

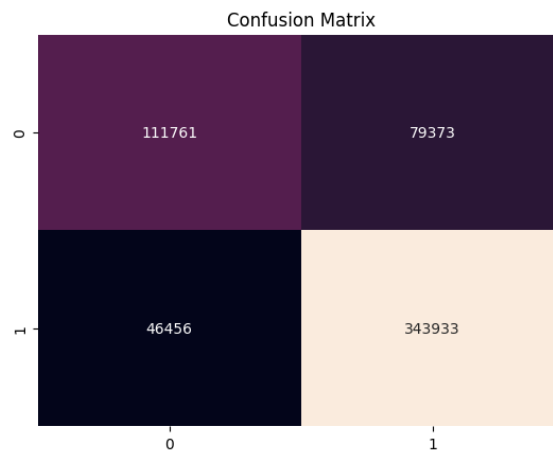
MODEL EVALUATION

Pada bab ini dijelaskan terkait evaluasi terhadap model kasus *Classification* Status Kepesertaan pada Data BPJS Kesehatan Tahun 2015-2021 menggunakan algoritma Random Forest *Classification*.

5.1 Evaluate Result

Tahap kelima merupakan tahap evaluasi terhadap model yang telah dirancang untuk melakukan klasifikasi dengan algoritma *random forest*. Tujuan dari evaluasi ini untuk melihat bahwa model telah mencapai hasil baik yang sesuai dengan standar pada tahap business understanding. Berikut ini adalah hasil evaluasi model yang didapatkan

- *Accuracy test* : 0.78
- *Precision test* : 0.81
- *Recall test* : 0.88



Gambar 17. Confussion Matrix

Pada proyek ini merupakan binary classification sehingga memiliki nilai [0,1] yang menandakan 0 berarti terdapat tidak aktif dan 1 terdapat aktif. Pada heatmap di atas diperoleh korelasi antara predicted table dengan true table untuk menghasilkan nilai valid dan tidak valid. Nilai valid merupakan nilai yang diprediksi tidak aktif dan benar tidak aktif berjumlah 111761 dan data yang diprediksi aktif dan benar aktif berjumlah 343933. Sedangkan data tidak valid adalah data yang diprediksi tidak aktif namun ternyata aktif berjumlah 79373 dan data yang diprediksi tidak aktif namun ternyata aktif adalah berjumlah 46456.

5.2 Evaluate Process

Tahap ini memeriksa kembali tahapan dari awal untuk memastikan bahwa tidak ada faktor penting dalam proses tersebut yang terabaikan atau terlewat. Berdasarkan hasil peninjauan proses awal proyek data mining dengan metodologi CRISP-DM, maka dapat dipahami bahwa:

- Proses eksplorasi data akan membantu dalam memilih atribut yang berkaitan dengan klasifikasi status kepesertaan pada layanan BPJS.
- Data Preparation, khususnya pada proses data cleaning dan transform, sehingga data yang diperoleh dapat menghasilkan model yang baik.
- Sangat penting untuk tetap fokus pada masalah bisnis yang dihadapi, karena setelah data siap dianalisis, maka akan dilakukan tahap pemodelan.

BAB 6

DEPLOYMENT

Tahap keenam pada metodologi CRISP-DM untuk melakukan klasifikasi status kepesertaan adalah deployment. Pada bab ini akan dijelaskan mengenai perencanaan dan deployment model yang sudah dihasilkan, serta laporan akhir untuk proses data mining yang sudah dilakukan. Dalam proses ini, kita akan menggunakan metode CRISP-DM untuk mengelola proses data mining secara sistematis dan menghasilkan hasil yang bermanfaat bagi pemangku kepentingan. Dengan mengikuti langkah-langkah dalam CRISP-DM, kita dapat memastikan keberhasilan proyek data mining dan membantu pemerintah dalam mengatasi masalah inefisiensi dalam sistem BPJS Kesehatan.

6.1 Membuat Rencana Deployment Model

Model *deployment* adalah proses di mana model yang telah dibangun akan tersedia pada lingkungan produksi, di mana model tersebut dapat melakukan prediksi pada sistem lain. Dalam proyek ini, model *deployment* akan dilakukan dengan menampilkan model dalam bentuk website melalui web browser. Salah satu cara untuk melakukan *deployment* model adalah dengan menggunakan *Visual Studio Code* yang beralamat di *LocalHost* untuk mengelola dan menjalankan dari model yang dikembangkan.

6.2 Melakukan Deployment Model

Melakukan deployment model klasifikasi kepesertaan pada data BPJS Kesehatan tahun 2015-2021 menggunakan algoritma Random Forest dengan Visual Studio Code adalah suatu proses yang memerlukan beberapa langkah penting.

Berikut adalah langkah-langkah dalam melakukan Deployment Model:

1. Pastikan Anda sudah menginstall *Anaconda*
2. Buka terminal/*command prompt/power shell*
3. Cek Ketersediaan *Environment*
conda info --envs
4. Jika tidak tersedia, nuat virtual environment dengan
conda create -n <nama-environment> python=3.9
4. Aktifkan virtual environment dengan

conda activate <nama-environment>

5. Install semua dependency/package Python dengan

pip install flask gunicorn scikit-learn==1.2.2

6. Jalankan API menggunakan perintah

python app.py

Berikut ini adalah isi dari perintah kode *app.py*

```
from flask import Flask, request, render_template
import pickle
import pandas as pd # Pastikan Anda mengimpor pandas

# Import fungsi label_encode_columns dari file yang sesuai
from utils import label_encode_columns, columns_to_encode # tambahkan
import columns_to_encode

app = Flask(__name__)

model_file = open('modeldtc.sav', 'rb')
model = pickle.load(model_file)

# Fungsi label_encode_columns yang telah Anda definisikan sebelumnya

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

@app.route('/predict', methods=['POST'])
def predict():
    PSTV02, PSTV04, PSTV05, PSTV06, PSTV07, PSTV08, PSTV09, PSTV10,
    PSTV11, PSTV12, PSTV13, PSTV14, PSTV15, PSTV16 = [x for x in
    request.form.values()]

    # Membuat list data dari input form
    data = [int(PSTV02), PSTV04, PSTV05, PSTV06, PSTV07, PSTV08,
    PSTV09, PSTV10, PSTV11, PSTV12, PSTV13, PSTV14, float(PSTV15),
    int(PSTV16)]

    # Membuat dataframe dari list data
    new_data = pd.DataFrame([data], columns=columns_to_encode)
```

```

# Melakukan label encoding pada new data menggunakan fungsi
label_encode_columns
new_data_encoded = label_encode_columns(new_data,
columns_to_encode)

# Melakukan prediksi menggunakan model
prediction = model.predict(new_data_encoded)

# Thresholding dengan nilai threshold 0.5
output = (prediction > 0.5).astype(int)

# Menyusun pesan berdasarkan output
if output == 1:
    message = "Aktif"
else:
    message = "Tidak Aktif"

return render_template('index.html', insurance_cost=message)

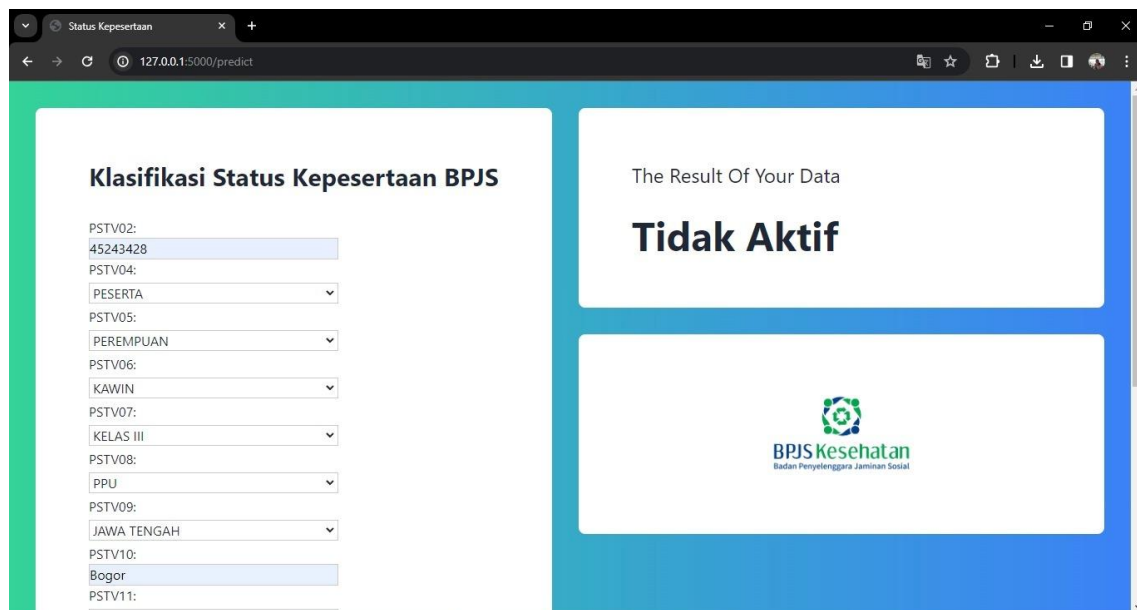
if __name__ == '__main__':
    app.run(debug=True)

```

Gambar 18. Kode Perintah app.py

7. Akses web di localhost:5000

8. Tampilan hasil deployment



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/predict'. The page has a teal and blue header. The main content area is divided into two sections. The left section, titled 'Klasifikasi Status Kepesertaan BPJS', contains a list of input fields for various data points (PSTV02 to PSTV11). The right section, titled 'The Result Of Your Data', displays the prediction result 'Tidak Aktif' in large, bold black text. Below the result, there is a logo for 'BPJS Kesehatan' (Badan Penyelenggara Jaminan Sosial).

Klasifikasi Status Kepesertaan BPJS

PSTV02: 45243428

PSTV04: PESERTA

PSTV05: PEREMPUAN

PSTV06: KAWIN

PSTV07: KELAS III

PSTV08: PPU

PSTV09: JAWA TENGAH

PSTV10: Bogor

PSTV11:

The Result Of Your Data

Tidak Aktif

BPJS Kesehatan
Badan Penyelenggara Jaminan Sosial

Gambar 19. Tampilan Hasil Deployment