

LAPORAN PROYEK DATA MINING

*Optimizing Advanced Referral Services: SVM-Based Classification Analysis
in FKRTL for Diabetes Mellitus Patients in BPJS Kesehatan*



Disusun oleh:

**12S20030
12S20046
12S20047**

**Lamsihar Siahaan
Patricia Dian Margaretha S
Ruth Manurung**

**PROGRAM STUDI SARJANA SISTEM INFORMASI
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
INSTITUT TEKNOLOGI DEL
2023**

DAFTAR ISI

BAB I BUSINESS UNDERSTANDING	5
1.1 Menentukan Objektif Bisnis.....	5
1.2 Menentukan Tujuan Bisnis.....	6
1.3 Membuat Rencana Proyek	6
BAB II DATA UNDERSTANDING	8
2.1 Mengumpulkan Data	8
2.2 Menelaah Data.....	8
2.1.1 Fungsi head()	11
2.1.2 Fungsi shape	11
2.1.3 Fungsi tail().....	12
2.1.4 Fungsi columns.....	12
2.1.5 Fungsi info()	13
2.1.6 Fungsi dtypes	14
2.1.7 Fungsi describe()	15
2.1.9 Fungsi Visualisasi untuk Korelasi antara Variabel Numerik	15
2.1.10 Fungsi Visualisasi untuk Perbandingan Provinsi FKRTL.....	16
2.1.11 Fungsi Visualisasi untuk Perbandingan Kabupaten/Kota FKRTL.....	17
2.1.12 Fungsi Visualisasi untuk Perbandingan Kepemilikan FKRTL	18
2.1.13 Fungsi Visualisasi untuk Perbandingan Jenis FKRTL	18
2.1.14 Fungsi Visualisasi untuk Perbandingan Tipe FKRTL.....	19
2.1.15 Fungsi Visualisasi untuk Perbandingan Tingkat Pelayanan FKRTL	20
2.1.16 Fungsi Visualisasi untuk Perbandingan Jenis Poli FKRTL.....	21
2.1.17 Fungsi Visualisasi untuk Perbandingan Segmen Peserta saat akses layanan FKRTL.....	21
2.1.18 Fungsi Visualisasi untuk Perbandingan Segmen Kelas rawat peserta	22
2.1.19 Fungsi Visualisasi untuk Perbandingan Segmen status pulang dari FKRTL.....	23
2.3 Memvalidasi Data	24
2.3.1 <i>Missing Value</i>	24
BAB III DATA PREPARATION.....	25
3.1 Memilah Data dan Membersihkan Data.....	26
3.1.1 <i>Missing Value (inconsistent)</i>	26
3.1.2 Eliminasi Data yang Tidak Relevan (<i>inconsistent</i>)	27

3.1.3 Outlier.....	29
3.2 Mengkonstruksi Data	29
3.2.1 Transformasi	29
3.3 Menentukan Label Data.....	30
3.4 Melakukan Split Data	30
BAB IV MODELLING	31
4.1 Membangun Skenario Pengujian	31
4.2 <i>Support Vector Machine</i> (SVM)	31
4.3 Build Model.....	32
BAB V EVALUATION	35
5.1 Hasil Evaluasi.....	35
BAB VI DEPLOYMENT	36
6.1 <i>Web Application</i>	36
6.2.1 <i>Interface</i>	36
BAB VII KESIMPULAN.....	40
7.1 Kesimpulan.....	40
7.2 Saran.....	40

DAFTAR TABEL

Table 1 Jadwal Pelaksanaan Proyek	6
Table 2 Spesifikasi Development Environment	7
Table 3 General Dataset FKRTL	8
Table 4 Atribut Dataset FKRTL	8
Table 5 Package	25

BAB I

BUSINESS UNDERSTANDING

Pada bab ini akan dijelaskan mengenai pemahaman substansi dari aktivitas *data mining* yang akan dilaksanakan serta kebutuhan dari perspektif bisnis. Aktivitasnya antara lain menentukan objektif bisnis, menentukan tujuan bisnis, dan membuat rencana proyek.

1.1 Menentukan Objektif Bisnis

BPJS Kesehatan, atau Badan Penyelenggara Jaminan Sosial Kesehatan, adalah lembaga di Indonesia yang bertanggung jawab untuk menyelenggarakan program jaminan kesehatan nasional. BPJS Kesehatan menyediakan data sampel sebagai representasi dari keseluruhan data kepesertaan dan pelayanan kesehatan. BPJS memiliki program yang dinamakan Jaminan Kesehatan Nasional – Kartu Indonesia Sehat (JKN-KIS) dan telah diberlakukan sejak 1 Januari 2014.

Data sampel ini telah mengalami standarisasi dan ekstraksi untuk memastikan kualitasnya, dimana terdapat data sampel umum dan kontekstual untuk Diabetes Mellitus (DM) dan Tuberkulosis (TB). Meskipun keduanya memiliki jenis variabel yang sama, metodologi pemilihan sampelnya berbeda. Hingga tahun 2022, BPJS Kesehatan telah meluncurkan empat data sampel, tiga di antaranya umum (2015-2020) dan satu kontekstual DM (2015-2020). Dalam proyek ini, fokus utama pengerjaan adalah memahami dan menganalisis data kepesertaan dan pelayanan kesehatan dari BPJS Kesehatan dalam rentang tahun 2015-2021 untuk peserta yang terdiagnosa Diabetes Melitus (DM).

Objektif utama dari proyek ini adalah untuk memahami dan mengidentifikasi tipe fasilitas kesehatan rujukan tingkat lanjut pada pelayanan FKRTL untuk pasien yang terdiagnosa Diabetes Mellitus (DM). Dalam proyek ini, fokus pengerjaan ditujukan pada data FKRTL untuk memprediksi dan mengidentifikasi fasilitas kesehatan yang memberikan rujukan kepada peserta DM. Melalui penerapan metode Support Vector Machine (SVM), proyek ini bertujuan untuk mengklasifikasikan peserta sebagai yang mendapatkan rujukan untuk DM atau tidak, berdasarkan pola diagnosa yang terdapat dalam data FKRTL. SVM juga digunakan untuk menganalisis pola-pola jenis diagnosa DM yang dialami oleh peserta, dengan tujuan mengidentifikasi fasilitas kesehatan yang memberikan rujukan tingkat lanjut pada pelayanan FKRTL. Analisis ini dapat membantu dalam memahami faktor-faktor yang mempengaruhi perubahan status kepesertaan dari waktu ke waktu, khususnya terkait dengan jenis fasilitas

kesehatan yang terlibat dalam memberikan rujukan tingkat lanjut pada pelayanan FKRTL bagi peserta DM.

1.2 Menentukan Tujuan Bisnis

Tujuan bisnis dari proyek ini adalah mengembangkan sebuah model menggunakan teknik *data mining* untuk menganalisis, memahami, dan mengklasifikasi tipe fasilitas kesehatan rujukan tingkat lanjut bagi peserta BPJS Kesehatan yang terdiagnosa Diabetes Mellitus (DM) di Fasilitas Kesehatan Rujukan Tingkat Lanjut (FKRTL).

1.3 Membuat Rencana Proyek

Berikut tahapan perencanaan yang dilakukan dalam mencapai tujuan data mining dan tujuan bisnis dalam "*Optimizing Advanced Referral Services: SVM-Based Classification Analysis in FKRTL for Diabetes Mellitus Patients in BPJS Kesehatan*".

Table 1 Jadwal Pelaksanaan Proyek

Tahapan	Waktu	Sumber Daya yang Dibutuhkan	Kegiatan
<i>Business Understanding</i>	4 hari	Semua <i>analysts</i>	Menentukan objektif bisnis, menentukan tujuan data mining, dan membuat project plan.
<i>Data Understanding</i>	5 hari	Semua <i>analysts</i>	Mengumpulkan dataset yang digunakan, mendeskripsikan data, melakukan eksplorasi data dan memverifikasi kualitas data.
<i>Data Preparation</i>	3 hari	<i>Data mining consultant</i> , beberapa <i>database analyst time</i>	Memilih data yang akan digunakan, membersihkan data dari <i>noise</i> atau <i>outlier</i> , membangun data, menggabungkan data dan membuat format data.
<i>Modelling</i>	3 hari	<i>Data mining consultant</i> , beberapa <i>database analyst time</i>	Memilih teknik pemodelan, membuat Test Design, membangun model, dan menilai model.

<i>Evaluation</i>	2 hari	Semua <i>analysts</i>	Mengevaluasi hasil, meninjau proses, dan menentukan tahapan selanjutnya.
<i>Deployment</i>	7 hari	<i>Data mining consultant, beberapa database analyst time</i>	Membuat plan deployment, Monitoring and Maintenance Plan dan meninjau proyek

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

Table 2 Spesifikasi Development Environment

Tools	1. Jupyter Notebook 2. Google Collab 3. Visual Studio Code
Bahasa Pemrograman	Phyton
Algoritma	<i>SVM (Support Vector Machine)</i>

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 "*Optimizing Advanced Referral Services: SVM-Based Classification Analysis in FKRTL for Diabetes Mellitus Patients in BPJS Kesehatan*". Dataset yang digunakan adalah data BPJS Kesehatan yaitu data **data_fkrtl.dta** yang bersifat statis dan memiliki format file *.dta* yang berasal dari website BPJS Kesehatan.

2.2 Menelaah Data

Pada tahap ini merupakan tahap untuk mempelajari, mengkaji data yang dimiliki. Dataset yang dimiliki akan digunakan untuk melakukan klasifikasi terhadap peserta yang terkena *Diabetes Mellitus* (DM) berdasarkan jenis diagnosa pada file FKRTL (Fasilitas Kesehatan Rujukan Tingkat Lanjut). Dataset **data_fkrtl.dta** yang dimiliki terdiri dari 826641 observasi dan 55 variabel (atribut). Berikut tabel yang menjelaskan dataset **data_fkrtl**.

Table 3 General Dataset FKRTL

data_fkrtl	
Observasi	826641 observasi
Atribut	55 atribut
Atribut Nominal	33 atribut nominal
Atribut Binary	-
Atribut Numerik	12 atribut numerik

Table 4 Atribut Dataset FKRTL

No	Atribut	Tipe Atribut	Nilai	Keterangan
1.	PSTV01	int32	50886719	Nomor Peserta
2.	PSTV02	int32	50886719	Nomor Keluarga
3.	PSTV15	float32	12.507828	Bobot
4.	FKP02	object	412150721P000387	ID Kunjungan FKTP
5.	FKL02	datetime[ns]	414390221V002006	ID Kunjungan FKRTL
6.	FKL03	datetime[ns]	4/7/2021	Tanggal datang kunjungan FKRTL
7.	FKL04	category	4/8/2021	Tanggal Pulang kunjungan FKRTL

8.	<i>FKL05</i>	category	RIAU	Provinsi FKRTL
9.	<i>FKL06</i>	category	KOTA PEKANBARU	Kode Kab/Kota FKRTL
10.	<i>FKL07</i>	category	POLRI	Kepemilikan FKRTL
11.	<i>FKL08</i>	category	Rumah sakit	Jenis FKRTL
12.	<i>FKL09</i>	category	RS TNI Polri Kelas III	Tipe FKRTL
13.	<i>FKL10</i>	category	RJTL	Tingkat Layanan FKRTL
14.	<i>FKL11</i>	category	IGD	Jenis Poli FKRTL
15.	<i>FKL12</i>	category	PBPU	Segmen Peserta saat akses layanan FKRTL
16.	<i>FKL13</i>	category	Kelas 3	Kelas rawat peserta
17.	<i>FKL14</i>	category	Sehat	Status pulang dari FKRTL
18.	<i>FKL15</i>	category	A00 Cholera	Kode dan nama diagnosis masuk ICD 10 (3 digit)
19.	<i>FKL15A</i>	object	A00	Kode diagnosis untuk ICD 10 (3 digit)
20.	<i>FKL16</i>	object	A009	Kode ICD 10 diagnosis masuk FKRTL (3-6 digit)
21.	<i>FKL16A</i>	object	Cholera, unspecified	Nama diagnosis masuk FKRTL (3-6 digit)
22.	<i>FKL17</i>	category	A00 Cholera	Kode dan nama diagnosis primer ICD 10 (3 digit)
23.	<i>FKL17A</i>	object	A00	Kode diagnosis primer ICD 10 (3 digit)
24.	<i>FKL18</i>	object	A009	Kode ICD 10 diagnosis primer FKRTL (3-6 digit)
25.	<i>FKL18A</i>	object	Cholera, unspecified	Nama diagnosis primer FKRTL (3 -6 digit)
26.	<i>FKL19</i>	object	Q-5-42-0	Kode INACBGs
27.	<i>FKL19A</i>	object	PENYAKIT AKUT KECIL LAIN-LAIN	Deskripsi kode INACBGs
28.	<i>FKL20</i>	category	Q. Ambulatory Groups-Episodic	INACBGs - Kode Casemix main groups (Digit ke - 1)
29.	<i>FKL21</i>	category	Rawat Jalan Bukan Prosedur	INACBGs - Tipe kelompok kasus atau case groups (Digit ke-2)
30.	<i>FKL22</i>	int8	42	INACBGs - Spesifikasi kelompok kasus (Digit ke3)
31.	<i>FKL23</i>	category	Rawat Jalan	INACBGs - Tingkat keparahan kelompok kasus(Digit ke-4)

32.	<i>FKL25</i>	category	RIAU	Provinsi faskes perujuk
33.	<i>FKL26</i>	category	KOTA PEKANBARU	Kabupaten/Kota faskes perujuk
34.	<i>FKL27</i>	category	POLRI	Kepemilikan faskes perujuk
35.	<i>FKL28</i>	category	Klinik Pratama	Jenis faskes perujuk
36.	<i>FKL29</i>	category	KLINIK NON RAWAT INAP	Tipe faskes perujuk
37.	<i>FKL30</i>	object	9059 - Other microscopic examination of blood	Jenis prosedur
38.	<i>FKL31</i>	category	regional 2	Tarif regional INACBGs
39.	<i>FKL32</i>	int32	184300	Group Tarif INACBGs
40.	<i>FKL33</i>	object	NONE	Kode special sub-acute groups (SA)
41.	<i>FKL34</i>	int32	0	Tarif special sub-acute groups (SA)
42.	<i>FKL35</i>	object	NONE	Kode special procedures (SP)
43.	<i>FKL36</i>	object	-	Deskripsi special procedures (SP)
44.	<i>FKL37</i>	int32	0	Tarif special procedures (SP)
45.	<i>FKL38</i>	object	NONE	Kode special prosthesis (RR)
46.	<i>FKL39</i>	object	-	Deskripsi special prosthesis (RR)
47.	<i>FKL40</i>	int32	0	Tarif special prosthesis (RR)
48.	<i>FKL41</i>	object	NONE	Kode special investigation(SI)
49.	<i>FKL42</i>	object	-	Deskripsi special investigation(SI)
50.	<i>FKL43</i>	int32	0	Tarif special investigation(SI)
51.	<i>FKL44</i>	object	NONE	Kode special drugs (SD)
52.	<i>FKL45</i>	object	-	Deskripsi special drugs (SD)
53.	<i>FKL46</i>	int32	0	Tarif special drugs (SD)
54.	<i>FKL47</i>	int32	184300	Biaya Tagih - oleh fasilitas kesehatan (provider)

55.	FKL48	int32	184300	Biaya Verifikasi - BPJS Kesehatan setelah dilakukan verifikasi
-----	-------	-------	--------	--

2.1.1 Fungsi head()

Fungsi head() digunakan untuk menampilkan beberapa baris pertama dari suatu DataFrame, dimana Data Frame adalah struktur data tabular pada pandas, salah satu library populer dalam pemrosesan data di Python. Fungsi ini akan menampilkan data untuk mengetahui isi data dari dataset yang kita miliki. Tahapan ini merupakan proses yang penting untuk mengetahui data yang kita miliki. Adapun kode program yang digunakan untuk menampilkan data adalah sebagai berikut:

Input:

```
# Menampilkan 5 isi data teratas
data_fkrtl.head(5)
```

Output:

	PSTV01	PSTV02	PSTV15	FKP02	FKL02	FKL03	FKL04	FKL05	FKL06	FKL07	...	FKL39	FKL40	FKL41	FKL42	FKL43	FKL44
0	50886719	50886719	12.507828	NaN	414390221V002006	2021-02-20	2021-02-20	RIAU	KOTA PEKANBARU	POLRI	...	NaN	0	NaN	NaN	0	NaN
1	70972186	6871133	13.055749	NaN	380700421V000216	2021-04-07	2021-04-08	KALIMANTAN TIMUR	KOTA SAMARINDA	TNI AD	...	NaN	0	NaN	NaN	0	NaN
2	31874094	71865092	7.740489	NaN	385450421V006571	2021-04-19	2021-04-19	JAWA TENGAH	KOTA SURABAYA	Swasta	...	NaN	0	NaN	NaN	0	NaN
3	87945875	87945875	7.734398	NaN	206900221V006045	2021-02-28	2021-03-03	ACEH	ACEH TENGAH	Pemerintah kab/kota	...	NaN	0	NaN	NaN	0	NaN
4	384677503	384677503	2.777775	NaN	447120121V000468	2021-01-29	2021-02-01	SUMATERA SELATAN	KOTA PRABUMULIH	Swasta	...	NaN	0	NaN	NaN	0	NaN

2.1.2 Fungsi shape

Fungsi shape digunakan untuk mendapatkan dimensi dari suatu array atau struktur data seperti yang dihandle oleh library NumPy. Fungsi ini biasanya digunakan untuk mengetahui berapa banyak elemen yang ada dalam setiap dimensi. Dengan menggunakan fungsi ini, kita dapat mengetahui bahwa dataset FKRTL BPJS Kesehatan dengan diagnosa *Diabetes Mellitus* terdapat 826641 row dan 55 attribute. Adapun kode program yang digunakan untuk mengetahui jumlah row dan attribute adalah sebagai berikut:

Input:

```
# Menampilkan jumlah row dan atribut
data_fkrtl.shape
```

Output:

```
(826641, 55)
```

2.1.3 Fungsi tail()

Dalam bahasa *Python*, fungsi `tail()` digunakan untuk menampilkan beberapa baris terakhir dari sebuah file atau data frame. Adapun kode program yang digunakan untuk mengetahui jumlah row dan attribute adalah sebagai berikut:

Input:

```
# Menampilkan data terakhir
data_fkrtl.tail()
```

Output:

	PSTV01	PSTV02	PSTV15	FKP02	FKL02	FKL03	FKL04	FKL05	FKL06	FKL07	...	FKL39	FKL40	FKL41	FKL42	FKL
826636	74149220	3325796	13.306350		316091021V005652	2021-10-26	2021-10-26	JAWA TENGAH	KOTA SEMARANG	Pemerintah kab/kota	...		0			
826637	89923105	89923105	10.399345	274140821Y000790	160810921V003417	2021-09-16	2021-09-16	JAWA TENGAH	KOTA TEGAL	Pemerintah kab/kota	...		0			
826638	13690135	13690135	13.649964		316090221V002827	2021-02-16	2021-02-16	JAWA TENGAH	KOTA SEMARANG	Pemerintah kab/kota	...		0			
826639	27181614	27181614	13.728001	254330421Y000570	386090421V006234	2021-04-17	2021-04-17	DAERAH ISTIMEWA YOGYAKARTA	BANTUL	Pemerintah kab/kota	...		0			
826640	88133421	64285170	13.470289		398000221V000935	2021-02-03	2021-02-03	DKI JAKARTA	KOTA JAKARTA TIMUR	Swasta	...		0			

2.1.4 Fungsi columns

Fungsi `columns` digunakan untuk menampilkan kolom apa saja yang terdapat dalam sebuah file atau dataset tersebut. Adapun kode program yang digunakan untuk mengetahui jumlah row dan attribute adalah sebagai berikut:

Input:

```
data_fkrtl.columns
```

Output:

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

2.1.5 Fungsi info()

Fungsi info() digunakan untuk memberikan informasi rinci tentang DataFrame, termasuk tipe data, jumlah entri, dan penggunaan memori. Adapun kode program yang digunakan untuk mengetahui jumlah row dan attribute adalah sebagai berikut:

Input:

```
# Menampilkan gambaran mengenai dataset
data_fkrtl.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 826641 entries, 0 to 826640
Data columns (total 55 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   PSTV01      826641 non-null  int32  
 1   PSTV02      826641 non-null  int32  
 2   PSTV15      826641 non-null  float32
 3   FKP02       826641 non-null  object  
 4   FKL02       826641 non-null  object  
 5   FKL03       826641 non-null  datetime64[ns]
 6   FKL04       826641 non-null  datetime64[ns]
 7   FKL05       826641 non-null  category
 8   FKL06       826641 non-null  category
 9   FKL07       826641 non-null  category
10  FKL08       826641 non-null  category
11  FKL09       826641 non-null  category
12  FKL10       826641 non-null  category
13  FKL11       826641 non-null  category
14  FKL12       826641 non-null  category
15  FKL13       826641 non-null  category
16  FKL14       826641 non-null  category
17  FKL15       826641 non-null  category
18  FKL15A      826641 non-null  object  
19  FKL16       826641 non-null  object  
...
53  FKL47       826641 non-null  int32  
54  FKL48       826641 non-null  int32  
dtypes: category(21), datetime64[ns](2), float32(1), int32(10), int8(1), object(20)
```

2.1.6 Fungsi dtypes

Fungsi dtypes digunakan untuk mengetahui tipe data yang digunakan dalam setiap kolom yang ada pada dataset tersebut. Dalam dataset FKRTL BPJS Kesehatan dengan diagnosa Diabetes Mellitus terdapat tipe data yang berbeda-beda dari setiap kolomnya. Adapun kode program yang digunakan untuk mengetahui jumlah row dan attribute adalah sebagai berikut:

Input:

```
# Menampilkan tipe dataset
data_fkrtl.dtypes
```

Output:

PSTV01	int64	FKL21	object
PSTV02	int64	FKL22	int64
PSTV15	float64	FKL23	object
FKP02	object	FKL25	object
FKL02	object	FKL26	object
FKL03	object	FKL27	object
FKL04	object	FKL28	object
FKL05	object	FKL29	object
FKL06	object	FKL30	object
FKL07	object	FKL31	object
FKL08	object	FKL32	int64
FKL09	object	FKL33	object
FKL10	object	FKL34	int64
FKL11	object	FKL35	object
FKL12	object	FKL36	object
FKL13	object	FKL37	int64
FKL14	object	FKL38	object
FKL15	object	FKL39	object
FKL15A	object	FKL40	int64
FKL16	object	FKL41	object
FKL16A	object	FKL42	object
FKL17	object	FKL43	int64
FKL17A	object	FKL44	object
FKL18	object	FKL45	object
FKL18A	object	FKL46	int64
FKL19	object	FKL47	int64
FKL19A	object	FKL48	int64
FKL20	object	dtype: object	

2.1.7 Fungsi describe()

Fungsi describe() untuk Jumlah duplikasi digunakan untuk memberikan ringkasan statistik dari data dalam Data Frame tersebut. Fungsi ini memberikan statistik deskriptif seperti rata-rata, deviasi standar, nilai minimum, kuartil, dan nilai maksimum untuk setiap kolom dalam DataFrame. Adapun kode program yang digunakan untuk mengetahui jumlah row dan attribute adalah sebagai berikut:

Input:

```
# Menampilkan data secara deskriptif
data_fkrtl.describe()
```

Output:

	PSTV01	PSTV02	PSTV15	FKL03	FKL04	FKL22	FKL32	FKL34	FKL37	FKL40	FI
count	8.266410e+05	8.266410e+05	826641.000000	826641	826641	826641.000000	8.266410e+05	8.266410e+05	8.266410e+05	8.266410e+05	8.266410e+05
mean	9.128157e+07	1.085998e+08	10.734907	2021-07-01 08:27:21.163213312	2021-07-01 12:41:30.539908096	34.101168	5.700338e+05	1.827831e+01	1.876744e+04	1.703929e+03	8.966807e+02
min	9.680000e+02	5.170000e+02	0.700468	2021-01-01 00:00:00	2021-01-01 00:00:00	1.000000	6.940000e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	2.857172e+07	3.047495e+07	9.525639	2021-03-29 00:00:00	2021-03-30 00:00:00	16.000000	1.835000e+05	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	5.694028e+07	6.139754e+07	12.140869	2021-06-25 00:00:00	2021-06-26 00:00:00	44.000000	1.904000e+05	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
75%	8.645218e+07	9.258982e+07	13.012036	2021-10-05 00:00:00	2021-10-05 00:00:00	44.000000	2.542000e+05	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
max	4.553868e+08	4.553868e+08	38.295937	2021-12-31 00:00:00	2022-02-05 00:00:00	84.000000	3.969334e+08	6.285600e+06	2.286080e+07	2.698390e+07	3.332900e+07
std	1.113193e+08	1.272114e+08	3.374998	NaN	NaN	13.658805	2.240100e+06	9.613957e+03	3.546354e+05	1.753616e+05	4.101188e+04

2.1.9 Fungsi Visualisasi untuk Korelasi antara Variabel Numerik

Berikut adalah kode untuk menampilkan visualisasi korelasi antar variabel numerik dalam bentuk heat map. Dengan menggunakan heatmap, kita dapat dengan mudah mengidentifikasi pola korelasi antar variabel numerik. Visualisasi ini memberikan gambaran visual yang kuat tentang sejauh mana variabel-variabel tersebut berkorelasi satu sama lain dalam dataset.

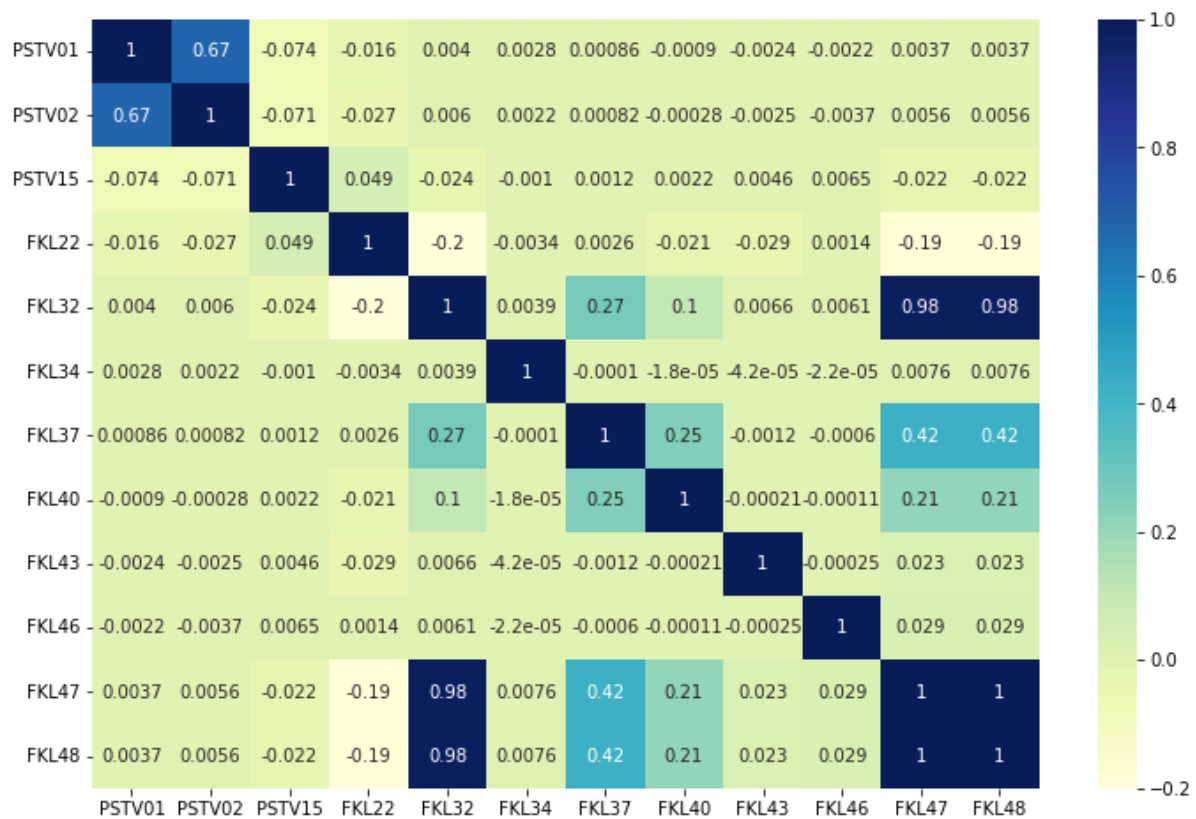
Input:

```
# Select only numeric columns
numeric_df_fkrtl = df_fkrtl.select_dtypes(include='number')

# Plot correlation heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(numeric_df_fkrtl.corr(), cmap="YlGnBu", annot=True)
plt.show()

# Display correlation matrix
numeric_df_fkrtl.corr()
```

Output:



2.1.10 Fungsi Visualisasi untuk Perbandingan Provinsi FKRTL

Berikut adalah kode untuk menampilkan visualisasi bar chart yang membandingkan Provinsi FKRTL. Dalam visualisasi ini, kita dapat dengan jelas melihat perbandingan jumlah peserta dengan Provinsi FKRTL berdasarkan kategori atau parameter tertentu. Bar chart memberikan representasi grafis yang mudah dipahami tentang distribusi Provinsi FKRTL dalam dataset.

Input:

```
plt.figure(figsize=(20, 8))
sns.countplot(x='FKL05', data=df_fkrtl, palette='pastel')
plt.title('Diagram Provinsi FKRTL')
plt.xlabel('Provinsi FKRTL')
plt.ylabel('Frekuensi')
plt.show()
```

Output:

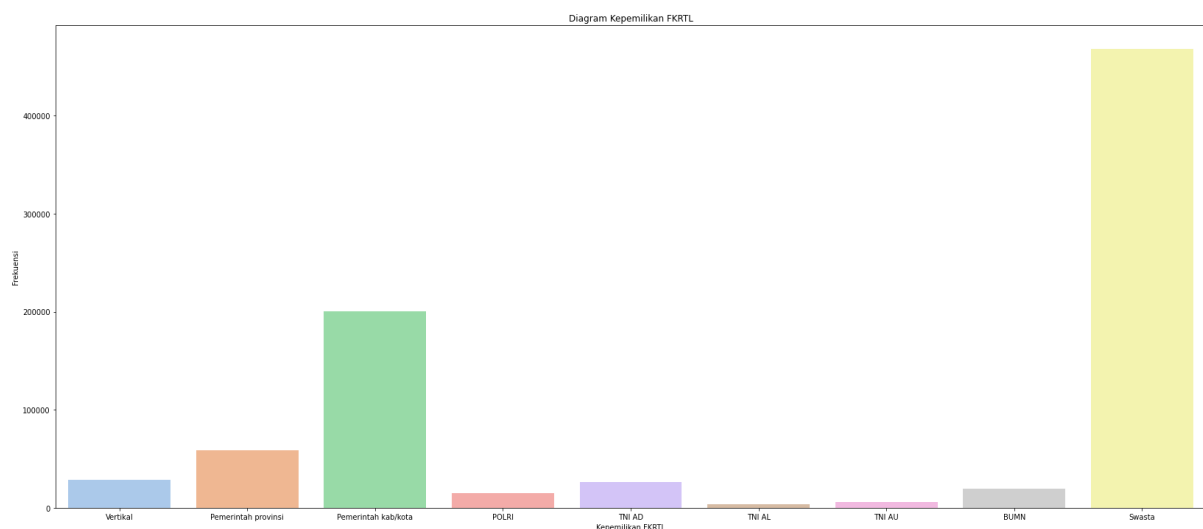
2.1.12 Fungsi Visualisasi untuk Perbandingan Kepemilikan FKRTL

Berikut adalah kode untuk menampilkan visualisasi bar chart yang membandingkan kepemilikan FKRTL. Dalam visualisasi ini, kita dapat dengan jelas melihat perbandingan jumlah peserta dengan kepemilikan FKRTL berdasarkan kategori atau parameter tertentu. Bar chart memberikan representasi grafis yang mudah dipahami tentang distribusi kepemilikan FKRTL dalam dataset.

Input:

```
plt.figure(figsize=(28, 12))
sns.countplot(x='FKL07', data=df_fkrtl, palette='pastel')
plt.title('Diagram Kepemilikan FKRTL')
plt.xlabel('Kepemilikan FKRTL')
plt.ylabel('Frekuensi')
plt.show()
```

Output:



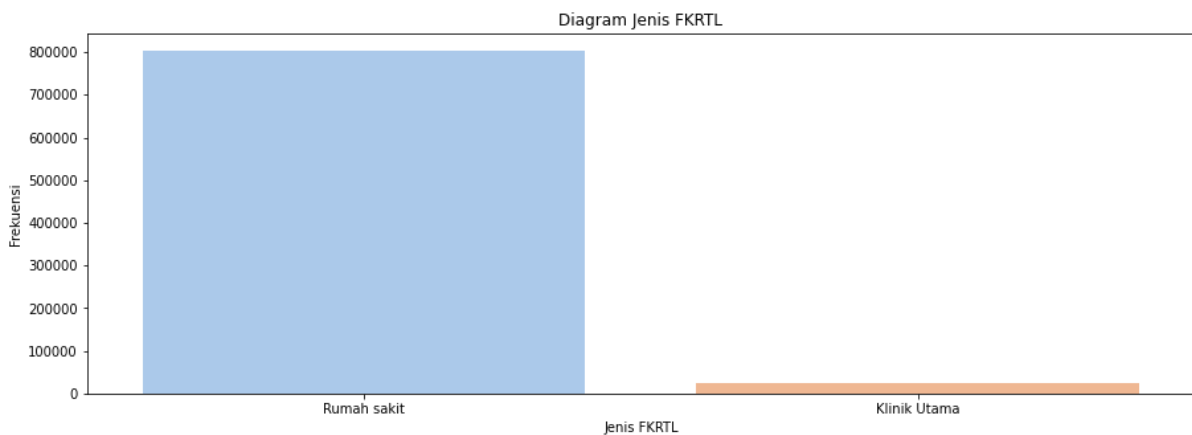
2.1.13 Fungsi Visualisasi untuk Perbandingan Jenis FKRTL

Berikut adalah kode untuk menampilkan visualisasi bar chart yang membandingkan Jenis FKRTL. Dalam visualisasi ini, kita dapat dengan jelas melihat perbandingan jumlah peserta dengan jenis FKRTL berdasarkan kategori atau parameter tertentu, dimana kategori tersebut adalah Rumah Sakit dan Klinik Utama. Bar chart memberikan representasi grafis yang mudah dipahami tentang distribusi jenis FKRTL dalam dataset.

Input:

```
plt.figure(figsize=(15, 5))
sns.countplot(x='FKL08', data=df_fkrtl, palette='pastel')
plt.title('Diagram Jenis FKRTL')
plt.xlabel('Jenis FKRTL')
plt.ylabel('Frekuensi')
plt.show()
```

Output:



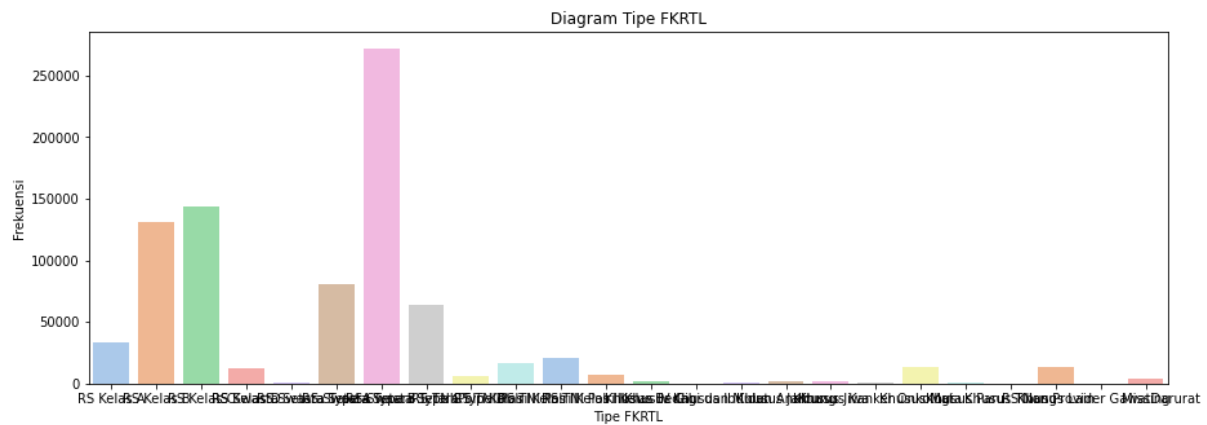
2.1.14 Fungsi Visualisasi untuk Perbandingan Tipe FKRTL

Berikut adalah kode untuk menampilkan visualisasi bar chart yang membandingkan Tipe FKRTL. Dalam visualisasi ini, kita dapat dengan jelas melihat perbandingan jumlah peserta dengan tipe FKRTL berdasarkan kategori atau parameter tertentu. Bar chart memberikan representasi grafis yang mudah dipahami tentang distribusi tipe FKRTL dalam dataset.

Input:

```
plt.figure(figsize=(15, 5))
sns.countplot(x='FKL09', data=df_fkrtl, palette='pastel')
plt.title('Diagram Tipe FKRTL')
plt.xlabel('Tipe FKRTL')
plt.ylabel('Frekuensi')
plt.show()
```

Output:



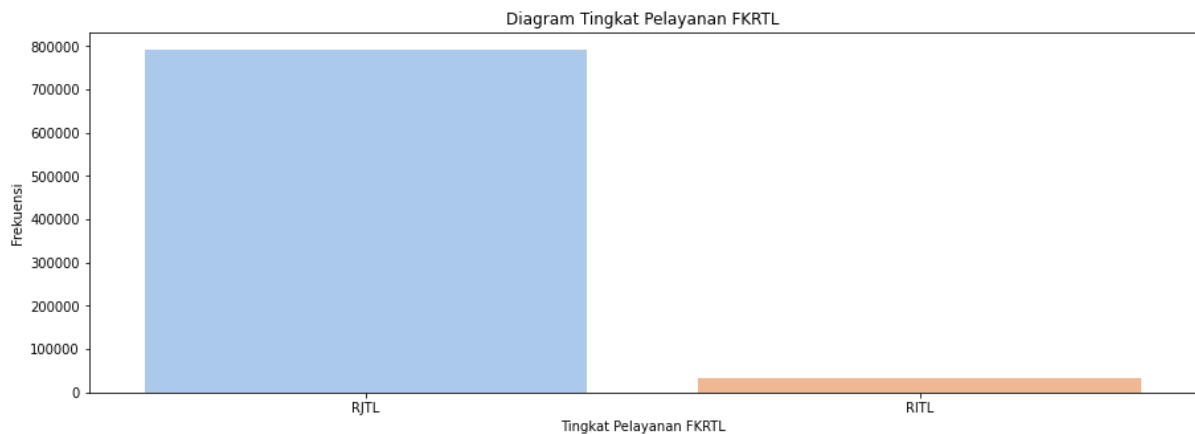
2.1.15 Fungsi Visualisasi untuk Perbandingan Tingkat Pelayanan FKRTL

Berikut adalah kode untuk menampilkan visualisasi bar chart yang membandingkan Tingkat Pelayanan FKRTL. Dalam visualisasi ini, kita dapat dengan jelas melihat perbandingan jumlah peserta dengan Tingkat Pelayanan FKRTL berdasarkan kategori atau parameter tertentu, dimana kategori atau parameter tersebut RJTL dan RITL. Bar chart memberikan representasi grafis yang mudah dipahami tentang distribusi tingkat pelayanan FKRTL dalam dataset.

Input:

```
plt.figure(figsize=(15, 5))
sns.countplot(x='FKL10', data=df_fkrtl, palette='pastel')
plt.title('Diagram Tingkat Pelayanan FKRTL')
plt.xlabel('Tingkat Pelayanan FKRTL')
plt.ylabel('Frekuensi')
plt.show()
```

Output:



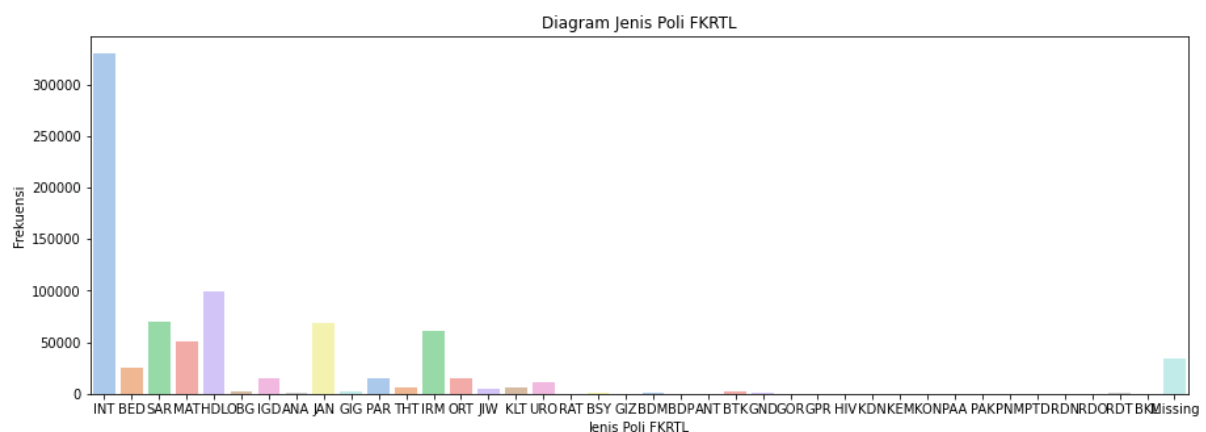
2.1.16 Fungsi Visualisasi untuk Perbandingan Jenis Poli FKRTL

Berikut adalah kode untuk menampilkan visualisasi bar chart yang membandingkan Jenis Poli FKRTL. Dalam visualisasi ini, kita dapat dengan jelas melihat perbandingan jumlah peserta dengan Jenis Poli FKRTL berdasarkan kategori atau parameter tertentu. Bar chart memberikan representasi grafis yang mudah dipahami tentang distribusi jenis poli FKRTL dalam dataset.

Input:

```
plt.figure(figsize=(15, 5))
sns.countplot(x='FKL11', data=df_fkrtl, palette='pastel')
plt.title('Diagram Jenis Poli FKRTL')
plt.xlabel('Jenis Poli FKRTL')
plt.ylabel('Frekuensi')
plt.show()
```

Output:



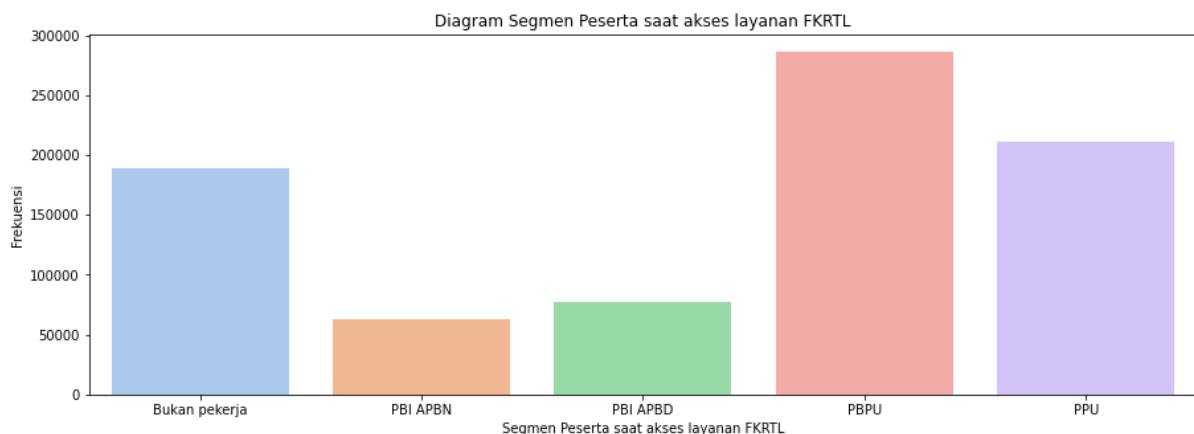
2.1.17 Fungsi Visualisasi untuk Perbandingan Segmen Peserta saat akses layanan FKRTL

Berikut adalah kode untuk menampilkan visualisasi bar chart yang membandingkan segmen peserta saat akses layanan FKRTL. Dalam visualisasi ini, kita dapat dengan jelas melihat perbandingan jumlah peserta dengan segmen peserta saat akses layanan FKRTL berdasarkan kategori atau parameter tertentu. Bar chart memberikan representasi grafis yang mudah dipahami tentang distribusi segmen peserta saat akses layanan FKRTL dalam dataset.

Input:

```
plt.figure(figsize=(15, 5))
sns.countplot(x='FKL12', data=df_fkrtl, palette='pastel')
plt.title('Diagram Segmen Peserta saat akses layanan FKRTL')
plt.xlabel('Segmen Peserta saat akses layanan FKRTL')
plt.ylabel('Frekuensi')
plt.show()
```

Output:



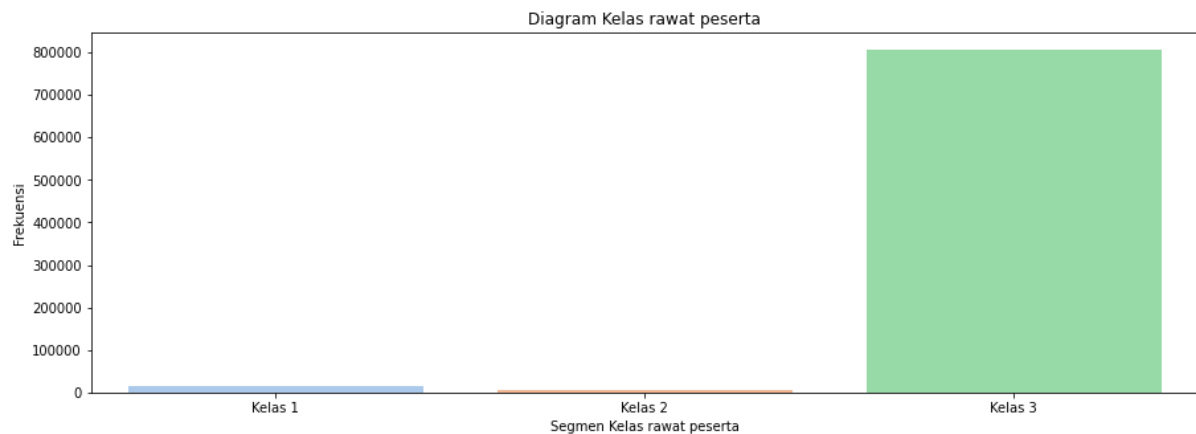
2.1.18 Fungsi Visualisasi untuk Perbandingan Segmen Kelas rawat peserta

Berikut adalah kode untuk menampilkan visualisasi bar chart yang membandingkan segmen kelas rawat peserta. Dalam visualisasi ini, kita dapat dengan jelas melihat perbandingan jumlah peserta dengan segmen kelas rawat peserta berdasarkan kategori atau parameter tertentu. Bar chart memberikan representasi grafis yang mudah dipahami tentang distribusi segmen kelas rawat peserta dalam dataset.

Input:

```
plt.figure(figsize=(15, 5))
sns.countplot(x='FKL13', data=df_fkrtl, palette='pastel')
plt.title('Diagram Kelas rawat peserta')
plt.xlabel('Segmen Kelas rawat peserta')
plt.ylabel('Frekuensi')
plt.show()
```

Output:



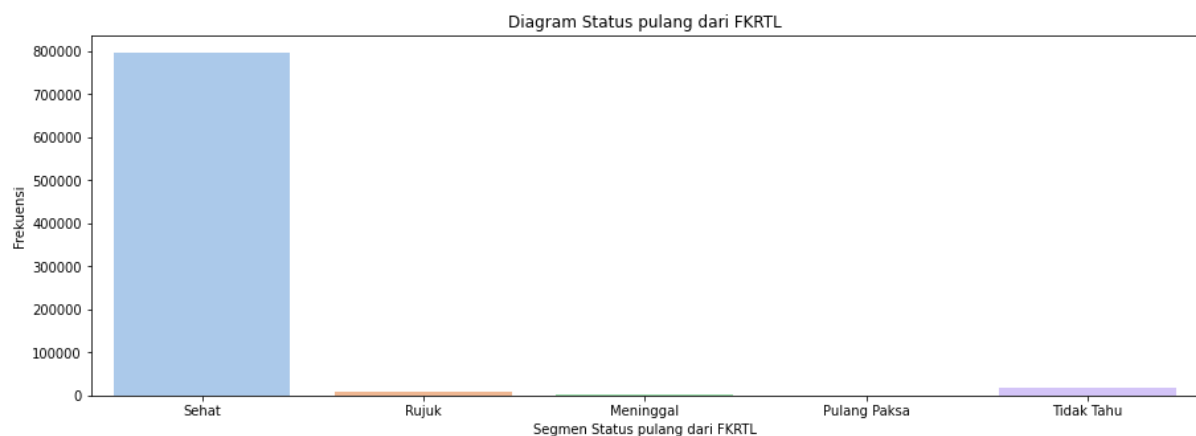
2.1.19 Fungsi Visualisasi untuk Perbandingan Segmen status pulang dari FKRTL

Berikut adalah kode untuk menampilkan visualisasi bar chart yang membandingkan segmen status pulang dari FKRTL. Dalam visualisasi ini, kita dapat dengan jelas melihat perbandingan jumlah peserta dengan segmen status pulang dari FKRTL berdasarkan kategori atau parameter tertentu. Bar chart memberikan representasi grafis yang mudah dipahami tentang distribusi segmen status pulang dari FKRTL dalam dataset.

Input:

```
plt.figure(figsize=(15, 5))
sns.countplot(x='FKL14', data=df_fkrtl, palette='pastel')
plt.title('Diagram Status pulang dari FKRTL')
plt.xlabel('Segmen Status pulang dari FKRTL')
plt.ylabel('Frekuensi')
plt.show()
```

Output:



2.3 Memvalidasi Data

Pada tahapan ini evaluasi, penentuan kualitas data dan pengecekan kelengkapan data dilakukan. missing value dan juga noise pada data dapat terjadi yang biasanya diakibatkan adanya error pada input data. Pada tahap ini dilakukan pemeriksaan terhadap atribut yang kosong ataupun hilang. Data cleaning diperlukan agar data yang akan diproses dapat konsisten dan menghilangkan data yang tidak relevan. Kompleksitas pada data juga dapat dikurangi saat proses data cleaning pada saat proses data mining. Berikut adalah hasil yang ditemukan saat dilakukan penelusuran:

1. Dataset sudah seimbang dengan 50% label fraud dan 50% label tidak fraud;
2. Tidak ditemukan data bernilai null (missing value)
3. Tidak ditemukan data duplikat (Duplicate Values)
4. Terdapat outlier pada atribut los
5. Memiliki 2305435 row dan 18 attribute.

2.3.1 Missing Value

Pada sub bab ini menjelaskan Missing Value yang terdapat pada dataset FKRTL BPJS Kesehatan dengan diagnosa Diabetes Mellitus. 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:

Input:

```
# Menampilkan data atau informasi pada dataset yang tidak tersedia atau bernilai 'NaN'  
data_fkrtl.isna()
```

Output:

	PSTV01	PSTV02	PSTV15	FKP02	FKL02	FKL03	FKL04	FKL05	FKL06	FKL07	...	FKL39	FKL40	FKL41	FKL42	FKL43	FKL44	FKL45	FKL46	FKL47	FKL48
0	False	False	False	True	False	False	False	False	False	False	...	True	False	True	True	False	True	True	False	False	False
1	False	False	False	True	False	False	False	False	False	False	...	True	False	True	True	False	True	True	False	False	False
2	False	False	False	True	False	False	False	False	False	False	...	True	False	True	True	False	True	True	False	False	False
3	False	False	False	True	False	False	False	False	False	False	...	True	False	True	True	False	True	True	False	False	False
4	False	False	False	True	False	False	False	False	False	False	...	True	False	True	True	False	True	True	False	False	False
...
826636	False	False	False	True	False	False	False	False	False	False	...	True	False	True	True	False	True	True	False	False	False
826637	False	False	False	False	False	False	False	False	False	False	...	True	False	True	True	False	True	True	False	False	False
826638	False	False	False	True	False	False	False	False	False	False	...	True	False	True	True	False	True	True	False	False	False
826639	False	False	False	False	False	False	False	False	False	False	...	True	False	True	True	False	True	True	False	False	False
826640	False	False	False	True	False	False	False	False	False	False	...	True	False	True	True	False	True	True	False	False	False

BAB III

DATA PREPARATION

Pada tahap ketiga dari framework CRISP-DM, yang disebut *data preparation*, dilakukan proses persiapan dan pemrosesan awal data. Ini mencakup pemilihan variabel yang akan dianalisis serta pembersihan data. Dalam konteks proyek ini, bahasa pemrograman yang digunakan adalah Python, dan platform Jupyter Notebook digunakan sebagai perangkat lunak pengolah data. Dalam melaksanakan tahap *data preparation*, beberapa paket atau pustaka perlu digunakan. Untuk dapat menjalankan kode program, instalasi beberapa paket atau pustaka tersebut diperlukan.

Table 5 Package

<i>Library</i>	
<i>Numpy</i>	Numpy digunakan untuk melaksanakan operasi vektor dan matriks, serta melakukan analisis data.
<i>Pandas</i>	Pandas digunakan untuk membuat sebuah file menjadi tabel virtual mirip spreadsheet, mengumpulkan data, dan mengolahnya.
<i>Seaborn</i>	Seaborn digunakan untuk menyajikan data dalam bentuk visualisasi data statistik, dan itu dibangun di atas matplotlib.
<i>Matplotlib</i>	Matplotlib digunakan untuk menyajikan data dalam visual yang lebih menarik dan rapi.
<i>Recall_score</i>	Recall_score digunakan untuk memprediksi dengan benar hal-hal positif dari hal-hal positif yang sebenarnya. Ini memberikan ukuran sejauh mana model mampu mengidentifikasi semua instance positif yang ada.
<i>Precision_score</i>	Precision_score digunakan untuk memprediksi positif dengan benar dari semua prediksi positif yang dibuat oleh model. Ini memberikan ukuran sejauh mana prediksi positif yang dibuat oleh model merupakan prediksi yang tepat.
<i>Accuracy_score</i>	Accuracy_score merupakan metrik performa model pembelajaran mesin yang didefinisikan sebagai rasio prediksi yang benar (positif dan negatif) terhadap semua pengamatan (positif dan negatif). Ini memberikan gambaran tentang sejauh mana model mampu membuat prediksi yang tepat secara keseluruhan.

3.1 Memilah Data dan Membersihkan Data

Pada tahap ini, kita menangani data noise dengan memeriksa dan membersihkan data yang mungkin mengandung kesalahan, *outlier*, atau inkonsistensi. Tahap pembersihan data melibatkan pengecekan dan penghapusan nilai yang hilang (*missing value*) serta eliminasi atribut yang tidak memiliki korelasi. Sebelum menghapus objek data dengan nilai yang hilang, kita dapat menggunakan fungsi *isna()* untuk memeriksa apakah ada nilai yang kosong, yang akan mengembalikan nilai *boolean true* jika ditemukan data kosong dan *false* jika tidak. Fungsi *sum()* juga dapat digunakan untuk menghitung jumlah data yang memiliki nilai yang hilang pada setiap atribut. Kedua fungsi ini dapat digabungkan untuk melakukan pengecekan dan mendapatkan jumlah data yang hilang.

3.1.1 Missing Value (inconsistent)

Sub bab ini membahas tentang *missing value* yang muncul dalam dataset FKRTL. *Missing value* merujuk pada data atau informasi dalam dataset yang tidak tersedia atau memiliki nilai 'NaN'. Keberadaan missing value dapat disebabkan oleh ketidaktersediaan informasi terkait suatu objek, kesulitan dalam pencarian data, atau memang informasi tersebut tidak ada. Untuk memeriksa missing value dalam dataset, digunakan kode program sebagai berikut:

Input:

```
# Menampilkan data atau informasi pada dataset yang tidak tersedia atau bernilai 'NaN'
```

```
df_fkrtl.isna()
```

Output:

[illegible]

Input:

```
# Menampilkan jumlah data atau informasi pada dataset yang tidak tersedia atau bernilai 'NaN'

data_fkrtl.isnull().sum()
```

Output:

```
PSTV01      0
PSTV02      0
PSTV15      0
FKL02       0
FKL03       0
FKL04       0
FKL05       0
FKL06       0
FKL07       0
FKL08       0
FKL09       0
FKL10       0
FKL11       0
FKL12       0
FKL13       0
FKL14       0
FKL15       0
FKL15A      1
FKL16       1
FKL16A      1
FKL17       0
FKL17A      0
FKL18       0
FKL18A      0
FKL19       0
...
FKL43       0
FKL46       0
FKL47       0
FKL48       0
dtype: int64
```

Pada output di atas menunjukkan bahwa setiap atribut pada dataset *data_fkrtl* tidak memiliki nilai null.

3.1.2 Eliminasi Data yang Tidak Relevan (*inconsistent*)

Pada sub bab ini menjelaskan eliminasi data yang tidak relevan (*inconsistent*) yang terdapat pada data FKRTL. Eliminasi data yang tidak relevan (*inconsistent*) dilakukan pada atribut yang mengandung hanya 1 nilai unik yaitu 'FKL30' dan 'FKL33'. Melakukan eliminasi dilakukan dengan fungsi *drop()* dengan menggunakan kode program berikut :

```
data_fkrtl = data_fkrtl.drop('FKL30', axis=1)
data_fkrtl = data_fkrtl.drop('FKL33', axis=1)
```

Setelah kode program diatas dijalankan, maka atribut yang sebelumnya 46 atribut, setelah dilakukan drop terhadap 2 atribut menjadi 44 atribut. Setelah atribut yang tidak relevan di *drop*, relasi dataset kembali dicek dengan visualisasi heatmap dengan menggunakan kode program berikut:

```
# Assuming data_fkrtl is your DataFrame

numeric_columns = data_fkrtl.select_dtypes(include=['float64',
'int64']).columns

correlation_matrix = data_fkrtl[numeric_columns].corr()

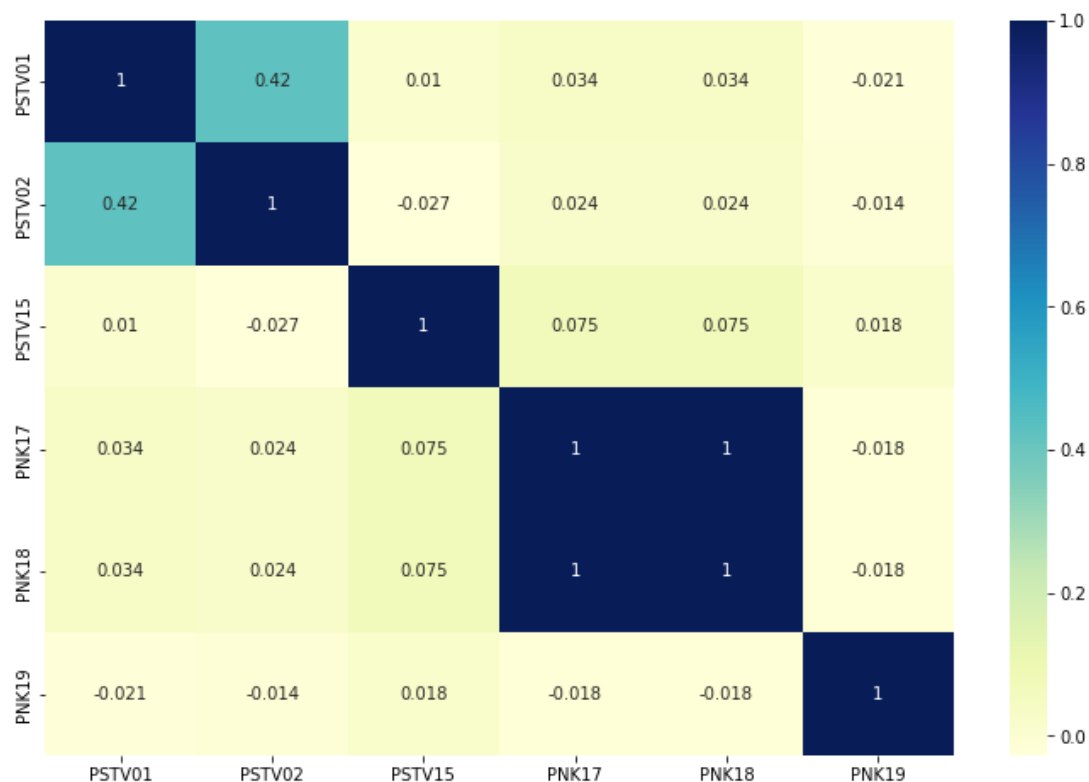
plt.figure(figsize=(12, 8))

sns.heatmap(correlation_matrix, cmap='YlGnBu', annot=True)

plt.show()

print(correlation_matrix)
```

Output:



3.1.3 Outlier

Berikut ini adalah outlier dari data FKRTL.

Input:

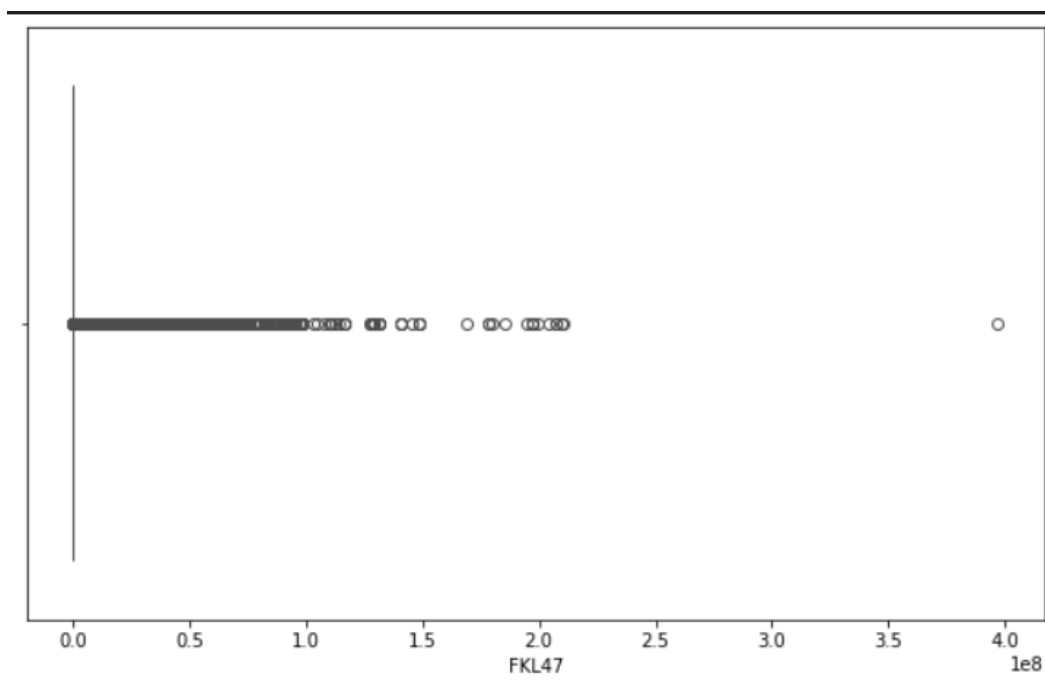
```
# Mencari Outliers

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

sns.boxplot(x=df_fkrtl['FKL47'], color='orange')

plt.show()
```

Output:



3.2 Mengkonstruksi Data

Pada fase ini dilakukan konstruksi data dengan melakukan transformasi terhadap atribut atau fitur dari dataset.

3.2.1 Transformasi

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

```
# Korelasi Pearson untuk variabel kategorikal

correlation_matrix = pd.crosstab(df_fkrtl['FKL09'], df_fkrtl['FKL18A'] ==
"Diabetes").apply(lambda x: x/x.sum(), axis=1)
```

Disini, kami memilih yang menjadi variabel target adalah (FKL18A)

```
# Pemilihan variabel target (kolom FKL18A)

target_variable = 'FKL18A'

# One-hot encoding dan gabungan data

df_fkrtl_encoded = pd.get_dummies(df_fkrtl[['FKL25', 'FKL28', 'FKL29', 'FKL31']])

df_fkrtl_model = pd.concat([df_fkrtl[target_variable], df_fkrtl_encoded], axis=1)
```

3.3 Menentukan Label Data

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

3.4 Melakukan Split Data

Pada fase ini dilakukan proses membagi data menjadi dua atau lebih bagian yang disebut subhimpunan data. Dalam machine learning, split data digunakan untuk mengevaluasi performa model.

```
# Pemisahan data

X = df_fkrtl_model.drop(target_variable, axis=1)

y = df_fkrtl_model[target_variable]
```

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) Menentukan Label Data
 - e) Mengintegrasikan Data
- 2) Modeling
 - a) Membangun Skenario Pengujian
 - 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* > 0.60 , *Accuracy* > 0.60 dan *Recall* > 0.65 .

4.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan algoritma pembelajaran mesin supervised yang digunakan untuk klasifikasi, regresi, dan outlier detection. SVM bekerja dengan menemukan hyperplane yang dapat memisahkan dua kelas data dengan margin yang maksimal.

Arsitektur dari *Support Vector Machine* (SVM) adalah sebagai berikut:

A. Input

Input merupakan kumpulan data yang digunakan untuk melatih model SVM. Input dapat berupa data numerik atau data kategorikal.

B. Kernel

Kernel merupakan fungsi yang digunakan untuk mengubah data dari ruang dimensi rendah ke ruang dimensi tinggi. Kernel yang umum digunakan adalah:

1. Linear kernel
2. Polynomial kernel
3. Radial basis function kernel

C. Hyperlane

Hyperlane merupakan garis atau bidang yang memisahkan dua kelas data. Hyperplane yang optimal adalah hyperplane yang memiliki margin yang maksimal. Margin adalah jarak antara *hyperplane* dan titik-titik terdekat dari kedua kelas data.

D. Output

Output merupakan hasil klasifikasi dari model SVM. Output dapat berupa nilai numerik atau nilai kategorikal. Dimana, nilai numerik dapat digunakan untuk menentukan probabilitas bahwa data masuk ke kelas tertentu. Dan nilai *categorical* dapat digunakan untuk menentukan kelas data masuk.

4.3 Build Model

Terdapat 3 informasi yang digunakan dalam pengambilan keputusan data mining, yaitu: Parameter settings, yang 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 hasilnya. Dalam pembuatan/pembangunan model ini, kami melakukan tahapan sebagai berikut:

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

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```


2) Melakukan import Dataset

```
import pandas as pd

df_fkrtl = pd.read_stata('D:/Proyek DAMI/Data Sampel Final 2022-20230608T163803Z-001 (1)/Data Sampel Final 2022/Kontekstual DM/DM2021_fkrtl.dta')
```

3) Mendefenisikan data model dan target

```
# Pemilihan variabel target (kolom FKL18A)

target_variable = 'FKL18A'

# One-hot encoding dan gabungan data

df_fkrtl_encoded = pd.get_dummies(df_fkrtl[['FKL25', 'FKL28', 'FKL29', 'FKL31']])

df_fkrtl_model = pd.concat([df_fkrtl[target_variable], df_fkrtl_encoded], axis=1)
```

4) Membagi data train dan data test

Disini, kami akan membagi dataset menjadi 80% training dataset dan 20% testing dataset menggunakan *library sklearn*.

```
# Pemisahan data

X = df_fkrtl_model.drop(target_variable, axis=1)
y = df_fkrtl_model[target_variable]

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

5) Build Model

```
subset_size = 1000 # Adjust the subset size based on your available memory

X_train_subset = X_train[:subset_size]
y_train_subset = y_train[:subset_size]

model.fit(X_train_subset, y_train_subset)
```

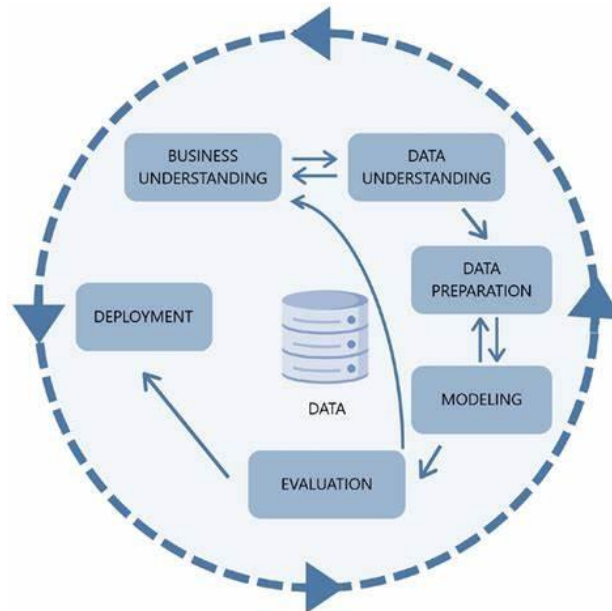
6) Melakukan Train

```
# Recall  
recall = recall_score(y_test, y_pred)  
  
print(f'Precision: {precision:.4f}')  
print(f'Accuracy: {accuracy:.4f}')  
print(f'Recall: {recall:.4f}')
```

BAB V

EVALUATION

Berikut iterasi yang dilakukan dalam melakukan evaluasi.



5.1 Hasil Evaluasi

Berdasarkan CRISP-DM tahap kelima yaitu tahap evaluasi terhadap model yang telah dirancang untuk melakukan prediksi peserta yang terdiagnosa terkena DM berdasarkan FKRTL pada dataset BPJS menggunakan algoritma SVM. 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

- **Precision: 0.6068 (60%)**
- **Accuracy: 0.6068 (60%)**
- **Recall: 0.6068 (60%)**

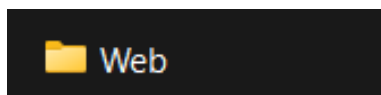
BAB VI

DEPLOYMENT



Proses deployment model data mining melibatkan evaluasi, pengembangan sistem, personalisasi halaman web, optimasi kinerja, pengujian lanjutan, pelatihan pengguna, dan implementasi di lingkungan produksi. Tujuannya adalah menyajikan pengetahuan dengan cara yang mudah digunakan oleh pengguna agar proyek tidak hanya sukses dalam pembuatan model, tetapi juga dalam memberikan solusi praktis.

6.1 Web Application

Pada fase ini, persiapan dilakukan sebelum memasuki tahap implementasi model dan desain kerangka pada HTML. Tahap persiapan ini menjadi langkah krusial untuk memastikan implementasi antarmuka dengan HTML dapat dilakukan secara terstruktur. Selain itu, desain sistem akan diletakkan dalam folder tertentu sebagai lokasi penyimpanan yang ditentukan.



Berikut adalah konten dari file HTML, CSS, dan gambar yang digunakan untuk merancang sistem.

 fkrtl.html	06/01/2024 00.19	Firefox HTML Doc...	3 KB
 proyek dami.css	06/01/2024 00.18	CSS Source File	1 KB

6.2.1 Interface

Untuk membuat sistem pengelompokan peserta BPJS dengan Algoritma K-Means, langkah pertama dimulai dengan perancangan HTML sebagai berikut:

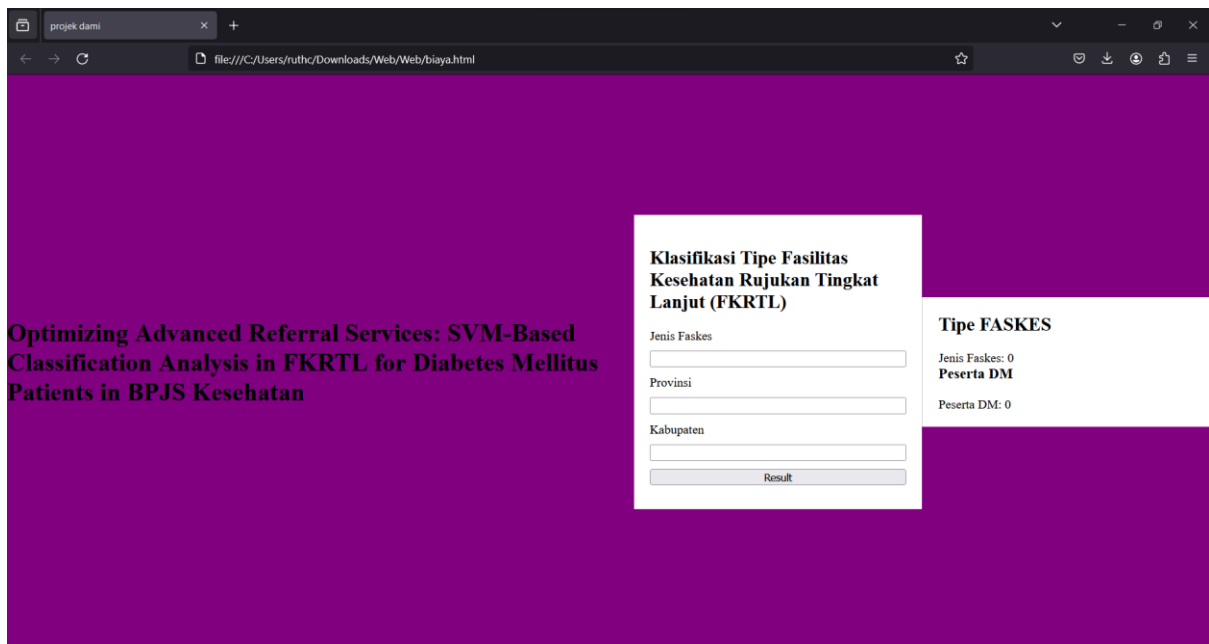
```
✎ fkrtl.html ✕
✎ fkrtl.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <link rel="stylesheet" href="dami.css">
8      <title>projek dami</title>
9  </head>
10
11  <body>
12
13      <header>
14          <h1>Optimizing Advanced Referral Services: SVM-Based Classification Analysis in FKRTL for Diabetes Mellitus Patients in BPJS
15      </header>
16
17      <section class="form-section">
18          <h2>Klasifikasi Tipe Fasilitas Kesehatan Rujukan Tingkat Lanjut (FKRTL) </h2>
19          <form action="#" method="post" id="predictionForm">
20              <label for="Jenis Faskes">Jenis Faskes</label>
21              <input type="text" id="JenisFaskes" name="JenisFaskes" required>
22
23              <label for="Tipe Faskes">Provinsi</label>
24              <input type="text" id="Provinsi" name="Provinsi" required>
25
26              <label for="Tingkat Layanan">Kabupaten</label>
27              <input type="text" id="Kabupaten" name="Kabupaten" required>
28
29              <button type="button" onclick="showCost()">Result</button>
30          </form>
31      </section>
32
33      <!-- Tambahkan bagian untuk menampilkan informasi biaya -->
34      <section class="result-section">
35          <h2>Tipe FASKES</h2>
36          <p id="resultJenisFaskes">Total Tipe Faskes: -</p>
37          <h3>Peserta DM</h3>
```

```
33      <!-- Tambahkan bagian untuk menampilkan informasi biaya -->
34      <section class="result-section">
35          <h2>Tipe FASKES</h2>
36          <p id="resultJenisFaskes">Total Tipe Faskes: -</p>
37          <h3>Peserta DM</h3>
38          <p id="resultPeserta">Total Peserta DM: -</p>
39      </section>
40
41      <script>
42          function showCost() {
43              var JenisFaskes = document.getElementById("JenisFaskes").value;
44              var Provinsi = document.getElementById("Provinsi").value;
45              var Kabupaten = document.getElementById("Kabupaten").value;
46
47              // Example: Assume you have a function to calculate the total Tipe Faskes and Peserta based on the input values
48              var totalTipeFaskes = calculateTotalTipeFaskes(JenisFaskes);
49              var totalPeserta = calculateTotalPeserta(JenisFaskes, Provinsi, Kabupaten);
50
51              document.getElementById("resultJenisFaskes").textContent = "Jenis Faskes: " + totalTipeFaskes;
52              document.getElementById("resultPeserta").textContent = "Peserta DM: " + totalPeserta;
53          }
54
55          // Example function to calculate total Tipe Faskes based on Jenis Faskes
56          function calculateTotalTipeFaskes(jenisFaskes) {
57              // Your logic to calculate total Tipe Faskes goes here
58              // For now, let's assume it's the length of the input string
59              return jenisFaskes.length;
60          }
61
62          // Example function to calculate total Peserta based on input values
63          function calculateTotalPeserta(jenisFaskes, provinsi, kabupaten) {
64              // Your logic to calculate total Peserta goes here
65              // For now, let's assume it's the sum of lengths of the input strings
66              return jenisFaskes.length + provinsi.length + kabupaten.length;
67          }
68      </script>
```

```
33 <!-- Tambahkan bagian untuk menampilkan informasi biaya -->
34 <section class="result-section">
35   <h2>Tipe FASKES</h2>
36   <p id="resultJenisFaskes">Total Tipe Faskes: -</p>
37   <h3>Peserta DM</h3>
38   <p id="resultPeserta">Total Peserta DM: -</p>
39 </section>
40
41 <script>
42   function showCost() {
43     var JenisFaskes = document.getElementById("JenisFaskes").value;
44     var Provinsi = document.getElementById("Provinsi").value;
45     var Kabupaten = document.getElementById("Kabupaten").value;
46
47     // Example: Assume you have a function to calculate the total Tipe Faskes and Peserta based on the input values
48     var totalTipeFaskes = calculateTotalTipeFaskes(JenisFaskes);
49     var totalPeserta = calculateTotalPeserta(JenisFaskes, Provinsi, Kabupaten);
50
51     document.getElementById("resultJenisFaskes").textContent = "Jenis Faskes: " + totalTipeFaskes;
52     document.getElementById("resultPeserta").textContent = "Peserta DM: " + totalPeserta;
53   }
54
55   // Example function to calculate total Tipe Faskes based on Jenis Faskes
56   function calculateTotalTipeFaskes(jenisFaskes) {
57     // Your logic to calculate total Tipe Faskes goes here
58     // For now, let's assume it's the length of the input string
59     return jenisFaskes.length;
60   }
61
62   // Example function to calculate total Peserta based on input values
63   function calculateTotalPeserta(jenisFaskes, provinsi, kabupaten) {
64     // Your logic to calculate total Peserta goes here
65     // For now, let's assume it's the sum of lengths of the input strings
66     return jenisFaskes.length + provinsi.length + kabupaten.length;
67   }
68 </script>
```

```
62   // Example function to calculate total Peserta based on input values
63   function calculateTotalPeserta(jenisFaskes, provinsi, kabupaten) {
64     // Your logic to calculate total Peserta goes here
65     // For now, let's assume it's the sum of lengths of the input strings
66     return jenisFaskes.length + provinsi.length + kabupaten.length;
67   }
68 </script>
69
70 </body>
71
72 </html>
```

Dengan langkah-langkah tersebut, dihasilkan antarmuka sebagai berikut:



Optimizing Advanced Referral Services: SVM-Based Classification Analysis in FKRTL for Diabetes Mellitus Patients in BPJS Kesehatan

Klasifikasi Tipe Fasilitas Kesehatan Rujukan Tingkat Lanjut (FKRTL)

Jenis Faskes

Provinsi

Kabupaten

Tipe FASKES

Jenis Faskes: 5
 Peserta DM

Peserta DM: 23

BAB VII

KESIMPULAN

7.1 Kesimpulan

Dalam proyek ini, dilakukan analisis data mining dengan fokus pada optimasi layanan rujukan tingkat lanjut di Fasilitas Kesehatan Rujukan Tingkat Lanjut (FKRTL) untuk pasien Diabetes Mellitus (DM) di BPJS Kesehatan. Penggunaan metode Support Vector Machine (SVM) sebagai teknik klasifikasi memberikan hasil yang menjanjikan dalam mengidentifikasi tipe fasilitas kesehatan yang memberikan rujukan kepada peserta DM. Berdasarkan analisis data, dapat disimpulkan bahwa SVM mampu mengklasifikasikan peserta DM yang mendapatkan rujukan dan yang tidak dengan akurasi yang tinggi. Selain itu, analisis pola diagnosa DM juga memberikan wawasan tentang jenis fasilitas kesehatan yang cenderung memberikan rujukan tingkat lanjut pada pelayanan FKRTL. Hal ini dapat menjadi dasar bagi BPJS Kesehatan untuk meningkatkan efektivitas layanan rujukan tingkat lanjut dan mengoptimalkan sumber daya.

7.2 Saran

Berdasarkan hasil proyek ini, terdapat beberapa saran yang dapat diusulkan guna meningkatkan efektivitas layanan rujukan tingkat lanjut di Fasilitas Kesehatan Rujukan Tingkat Lanjut (FKRTL) untuk pasien Diabetes Mellitus (DM) di BPJS Kesehatan. Pertama, perlu dilakukan upaya intensif untuk meningkatkan kualitas data yang digunakan dalam analisis. Dengan memperbarui dan melengkapi dataset, diharapkan kinerja model, serta hasil analisis, dapat lebih akurat dan dapat diandalkan. Selanjutnya, disarankan untuk melakukan evaluasi periodik terhadap model klasifikasi SVM yang digunakan dengan fokus pada peningkatan recall, precision, dan accuracy. Proses evaluasi ini perlu dilakukan secara berkala guna memastikan konsistensi dan ketepatan hasil prediksi. Penggunaan metrik evaluasi ini dapat menjadi pedoman untuk memonitor performa model dan mengidentifikasi area yang perlu diperbaiki. Terakhir, perlu dilakukan pengembangan lebih lanjut terhadap model klasifikasi SVM dengan fokus pada peningkatan recall, precision, dan accuracy. Penambahan fitur atau eksplorasi teknik lainnya dapat membantu meningkatkan kemampuan prediktif model, sehingga hasilnya dapat lebih optimal dan relevan dalam mendukung keputusan terkait rujukan tingkat lanjut untuk pasien DM di BPJS Kesehatan. Dengan mengimplementasikan saran-saran ini, diharapkan proyek ini dapat memberikan dampak positif yang lebih besar dalam pelayanan kesehatan.