DOKUMEN PROYEK

12S14054 - PENAMBANGAN DATA

Fraud Detection with Support Vector Machine

Disusun oleh:

12S19015 Putri Anyelir BR Tobing

12S19016 Tiar Saroha Simamora

12S19048 Fitri Putri Sitorus

12S19059 Rut Yana Gultom

PROGRAM STUDI SARJANA SISTEM INFORMASI

FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO

INSTITUT TEKNOLOGI DEL

DESEMBER 2022

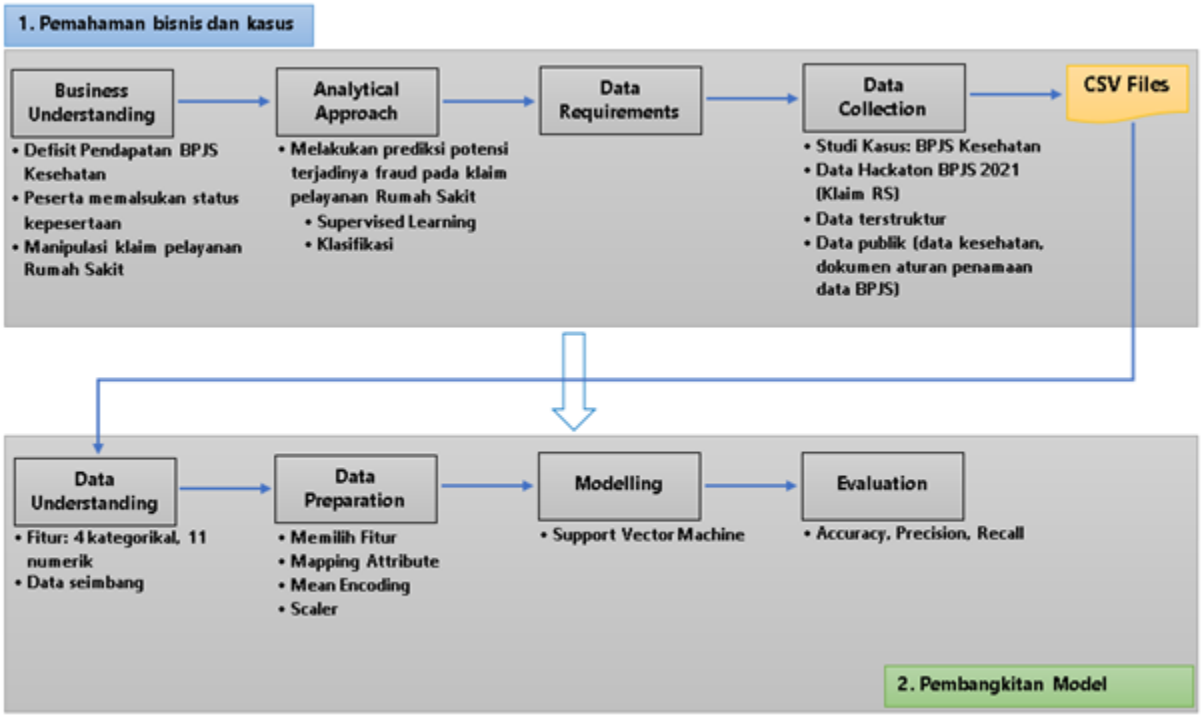
DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1 BUSINESS UNDERSTANDING

Tahap pertama proses CRISP-DM *Fraud Detection* adalah *business understanding*. Tahap ini akan menjelaskan aktivitas menentukan objektif bisnis, Menentukan tujuan bisnis, Membuat rencana proyek dan kebutuhan dari sudut pandang bisnis yang akan diartikan ke dalam pendefinisian masalah dalam *data mining*. Gambar 1 berikut ini menjelaskan tahapan berdasarkan metodologi CRISP-DM.



Gambar 1 Tahapan berdasarkan metodologi CRISP-DM

Tahap pemahaman bisnis dan studi kasus terdiri atas 4 proses yaitu:

1. Memahami bisnis atau studi kasus (*Business Understanding*)
2. Pendekatan analisis (*Analytical Approach*)
3. Kebutuhan data (*Data Requirements*)
4. Pengumpulan data (*Data Collection*)

Pemahaman masalah bisnis akan menghasilkan masalah untuk mencapai tujuan atau solusi. Berdasarkan masalah yang ditemukan, dilakukan analisis untuk memahami tujuan dan mengidentifikasi kebutuhan dari data mining yang akan dilakukan.

Permasalahan yang timbul pada *Fraud Detection with Support Vector Machine*, analisis dilakukan untuk mengembangkan sebuah model untuk melakukan prediksi potensi terjadinya fraud pada klaim pelayanan rumah sakit berdasarkan dataset yang ada.

1.1 Determine Business Objective

Di seluruh dunia, dalam konteks Jaminan Kesehatan Nasional, terdapat satu lembaga pembayaran dibentuk oleh Pemerintah yang bersangkutan, misalnya Badan Penyelenggara Jaminan Kesehatan (BPJS Kesehatan) yang mengelola Program Jaminan Kesehatan Nasional. BPJS Kesehatan yang sudah berdiri hampir satu dekade namun masih memiliki proses penjaminan yang lemah bagi peserta. Selain itu, kecurangan yang bisa dilakukan oleh petugas BPJS Kesehatan yakni melakukan kerjasama dengan peserta dan/atau fasilitas kesehatan untuk memanipulasi manfaat yang seharusnya tidak dijamin agar dapat dijamin, menahan pembayaran ke fasilitas kesehatan/rekanan dengan tujuan memperoleh keuntungan pribadi, membayar dana kapitasi tidak sesuai dengan ketentuan.

Tindakan-tindakan tersebut berdasarkan peraturan pemerintah termasuk pada tindakan kecurangan (*fraud*). *Fraud* mengacu pada tindakan tidak jujur yang dengan sengaja menggunakan penipuan untuk secara ilegal merampas uang, properti, atau hak hukum orang atau entitas lain. Berdasarkan faktor-faktor terjadinya *fraud* pada pelayanan kesehatan (BPJS) maka objektif yang akan dicapai dari proyek ini adalah melakukan prediksi potensi terjadinya *fraud* pada klaim pelayanan Rumah Sakit berdasarkan *dataset train* yang terdiri dari 200217 observasi dan 53 variabel. Proyek ini akan dikatakan sukses jika berhasil melakukan prediksi potensi terjadinya *fraud* pada klaim pelayanan Rumah Sakit.

1.2 *Situation Assessment*

Pengerjaan proyek ini menggunakan sumber daya berupa *hardware, data sources,* dan *personel. Hardware* yang digunakan adalah Asus VivoBook A412FL dan Lenovo ideapad 330 dengan i5-8265U *microprocessor* dan 8GB RAM. *Dataset* yang akan dianalisis pada pengerjaan proyek ini adalah data BPJS Hackathon . Data yang digunakan merupakan data yang bersifat status dikarenakan data memiliki format CSV (*Comma Separated Values*). Pengerjaan proyek ini akan melibatkan empat mahasiswa dalam setiap tahapan proyek yang dilakukan secara luring dengan kurun waktu kurang lebih 4 minggu. Dalam upaya memaksimalkan hasil pengerjaan proyek maka

dibutuhkan *management* waktu yang baik. Dalam pengerjaan proyek perlu diperhatikan pada *Cost/Benefit analysis*. Estimasi biaya yang diperlukan mencakup pengumpulan data dan biaya operasi (berupa biaya akses internet). Sementara itu benefit yang diperoleh adalah mampu mencapai 3 objektif utama, menambah wawasan yang diperoleh dari pemahaman data maupun pemahaman dalam tahapan pengerjaan proyek.

1.3 Determine Data Mining Goal

Tujuan bisnis pada proyek ini adalah untuk melakukan prediksi potensi terjadinya kecurangan (*fraud*) pada klaim pelayanan Rumah Sakit, ketika memprediksi kecurangan pada kalim dibutuhkan teknis penilaian yang efektif untuk memperoleh keputusan secara subjektif. Pada proyek ini menerapkan model untuk mengidentifikasi pola hubungan antar data, dimana pada model ini menggunakan metode *data mining.* Salah satu metode pada *data mining* adalah *support vector Machine* (SVM). Pendekatan ini didasarkan pada gagasan hyperplane classifier. Tujuan dari Support Vector Machine adalah untuk menemukan hyperplane optimal linier sehingga margin pemisahan antara dua kelas dimaksimalkan.

Pada penelitian yang dilakukan oleh Nana Kwame Gyamfi dan Dr Jamal-Deen Abdulai, pada tahun 2019 yang berjudul "Bank Fraud Detection Using Support Vector Machine". Penelitian tersebut menggunakan Support Vector Machine untuk membangun model yang mewakili perilaku pelanggan normal dan abnormal dan kemudian menggunakannya untuk mengevaluasi validitas transaksi baru. Hasil penelitian menunjukkan bahwa teknik SVM efektif dalam memerangi penipuan perbankan dalam data besar serta mencapai precision sekitar 80% .

Penelitian lainnya terkait Fraud Detection yaitu penelitian yang dilakukan oleh V.Dheepa and R. Dhanapala yang berjudul "*Behavior Based Credit Card Fraud Detection Using Support Vector Machine*s". Penelitian ini menggunakan metode SVM dan metode *feature extraction* untuk mendeteksi fraud yang terjadi. Jika terjadi ketidaksesuaian pada pola transaksi perilaku maka hal tersebut diduga mencurigakan dan akan menjadi bahan pertimbangan lebih lanjut untuk menemukan kecurangan tersebut. Metode yang digunakan memberikan akurasi deteksi yang lebih tinggi dan juga scalable untuk menangani volume transaksi yang besar.

Berdasarkan penelitian-penelitian *Fraud Detection* tersebut, metode *Support Vector Machine* dapat menangani hubungan antar item untuk melakukan prediksi fraud, maka penelitian ini akan

dilakukan dengan algoritma *Support Vector Machine* yang diharapkan menghasilkan prediksi yang akurat terhadap penelitian "*Fraud Detection using Support Vector Machine*" ini. Hasil keluaran menggunakan algoritma *Support Vector Machine* ini diharapkan dapat memenuhi minimum precision 60%, minimum accuracy 60%, dan minimum recall 65 %.

1.4 *Produce Project Plan*

Tabel 1 *Project Plan*

Tahapan	**Waktu**	**Sumber daya yang dibutuhkan**	**Kegiatan**	**Ketergantungan**
Business Understanding	3 days	*All analysts*	Menentukan tujuan utama bisnis, melakukan penilaian terhadap situasi, menentukan tujuan *data mining*, dan membuat *project plan*.	Perkembangan penerapan konsep *data mining*
Data understanding	4 days	*All analysts*	Mengumpulkan data yang akan digunakan, mendeskripsikan data, melakukan eksplorasi data, dan memverifikasi kualitas data.	Masalah data dan teknologi
Data preparation	5 days	*Data mining consultant, some database analyst time*	Memilih data yang akan digunakan, membersihkan data dari noise atau outlier, membangun data, menggabungkan data, dan membuat format data.	Masalah data dan teknologi
Modelling	4 days	*Data mining consultant, some database analyst time*	Memilih teknik pemodelan, membuat *Test Design*, membangun model, dan menilai model	Ketidakmampuan menemukan model yang tepat

Tahapan	**Waktu**	**Sumber daya yang dibutuhkan**	**Kegiatan**	**Ketergantungan**
Evaluation	2 days	*All analysts*	Mengevaluasi hasil, meninjau proses, dan menentukan tahapan selanjutnya	Ketidakmampuan untuk menerapkan hasil, terjadi kesalahan pada proses pengerjaan proyek, perkembangan penerapan konsep *data mining*

BAB 2 DATA UNDERSTANDING

Tahap kedua proses CRISP-DM *Fraud Detection* adalah *Data Understanding*.Pada Bab ini akan dijelaskan tahapan data understanding yaitu mengumpulkan data (*collecting data)*, deskripsi data untuk mendapatkan pemahaman tentang data yang akan digunakan dalam penelitian serta mengidentifikasi sebuah masalah kualitas data.

2.1 Collect Initial Data

Dalam tahapan data understanding, hal yang pertama dilakukan yaitu mengumpulkan data. Dimana hal ini, merupakan langkah persiapan untuk menemukan data awal. Data yang akan digunakan berasal dari dataset BPJS Hackathon. Dataset yang digunakan memiliki format file CSV (Comma Separated Values) sehingga datanya bersifat statis.

2.2 Describe Data

Dataset yang akan digunakan pada proyek ini yaitu untuk melakukan prediksi potensi terjadinya *fraud* pada klaim pelayanan Rumah Sakit berdasarkan *dataset train* yang terdiri dari 200217 observasi dan 53 atribut. Untuk atribut yang diawali dengan dx2 (diagnosa sekunder) yang terdiri dari 22 atribut yaitu: dx2_a00_b99, dx2_c00_d48, dx2_d50_d89, dx2_e00_e90, dx2_f00_f99, dx2_g00_g99, dx2_h00_h59, dx2_h60_h95, dx2_i00_i99, dx2_j00_j99, dx2_koo_k93, dx2_l00_l99, dx2_m00_m99, dx2_n00_n99, dx2_o00_o99, dx2_p00_p96, dx2_q00_q99, dx2_r00_r99, dx2_s00_t98, dx2_u00_u99, dx2_v01_y98, dx2_z00_z99 akan dikelompokkan menjadi satu atribut yaitu dx.

Kemudian untuk atribut yang diawali dengan proc (kode kelompok *procedure*) yang terdiri dari 19 atribut yaitu proc00_13, proc14_23, proc24_27, proc28_28, proc29_31, proc_32_38, proc39_45, proc46_51, proc52_57, proc58_62, proc63_67, proc68_70, proc71_73, proc74_75, proc76_77, proc78_79, proc80_99, proce00_e99, procv00_v89 akan dikelompokkan menjadi satu atribut yaitu proc. Berikut tabel untuk menjelaskan setiap atributnya.

Tabel 2 *Describe Data*

No.	Nama atribut	Tipe	Deskripsi
1.	visit_id	*int64*	id dari setiap kunjungan (unik)

No.	Nama atribut	Tipe	Deskripsi
2.	kdkc	*int64*	kode wilayah kantor cabang BPJS Kesehatan
3.	dati2	*int64*	kode kabupaten/kota
4.	typeppk	*object*	kode tipe Rumah Sakit
5.	jkpst	*object*	jenis kelamin peserta JKN-KIS
6.	umur	*int64*	umur peserta saat mendapatkan pelayanan rumah sakit
7.	jnspelsep	*int64*	tingkat pelayanan; 1:rawat inap; 2. rawat jalan
8.	los	*int64*	lama peserta dirawat di rumah sakit
9.	cmg	*object*	klasifikasi CMG (Case Mix Group)
10.	severitylevel	*int64*	tingkat urgensi
11.	diagprimer	*object*	diagnosa primer
12.	dx	*int64*	diagnosa sekunder
13.	proc	*int64*	kode kelompok *procedure*
14.	label	*int64*	*flag fraud*; 1: *fraud*; 0: tidak *fraud*

2.3. *Explore Data Analysis*

Exploratory Data Analysis (EDA) merupakan salah satu strategi yang digunakan untuk membantu dalam tahap pra pemrosesan penambangan data. *Exploratory Data* Analysis (EDA) bertujuan untuk mengkaji dan memahami dataset serta menyimpulkan karakteristik utamanya, seringkali memakai metode visualisasi data. Pendekatan ini digunakan untuk memahami data, mendapatkan

konteks data, memahami variabel dan hubungan di antara variabel, dan merumuskan hipotesis yang dapat berguna ketika membangun model prediksi. Atribut pada proyek ini merupakan atribut yang relevan dan sesuai dengan tujuan proyek yaitu atribut diagprimer. Beberapa hipotesis atribut yang relevan dengan atribut lainnya untuk memprediksi klaim palsu BPJS akan dijelaskan berikut ini:

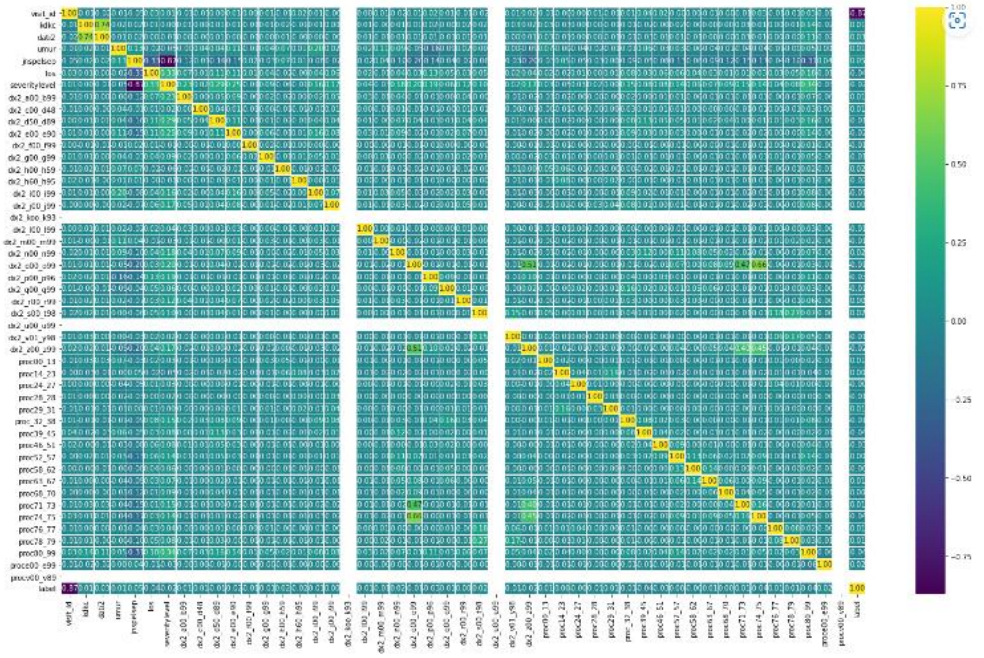
1. *Diagprimer* (*diagnosa primer*) berpengaruh terhadap prediksi klaim palsu BPJS, dimana kode diagnosa dan pelayanan dibuat lebih kompleks dari yang sebenarnya.
2. *Los* (lama peserta dirawat di rumah sakit) berpengaruh terhadap prediksi klaim palsu BPJS, dimana RS membuat suatu tagihan yang waktu rawatnya sebenarnya tidak sesuai dengan waktu yang sebenarnya.
3. *Procedure* (kode kelompok *procedure*) berpengaruh terhadap prediksi klaim palsu BPJS, dimana RS memasukkan klaim penagihan atas kode yang tidak akurat yaitu kode *procedure* yang lebih kompleks sehingga menghasilkan nilai klaim yang lebih tinggi dari yang seharusnya.

Dari beberapa hipotesis yang dirumuskan, terdapat beberapa atribut yang relevan yang selanjutnya akan digunakan untuk menentukan kecurangan klaim, yaitu variabel *diagprimer, los, dan procedure* berpengaruh terhadap kecurangan klaim tersebut.

2.3.1 Correlation

Karena semua atribut pada dataset bertipe data numerik, maka untuk melihat hubungan antara dua variabel digunakan *correlation*. Untuk menemukan korelasi antara variabel maka digunakan sebuah *heatmap* yang akan memvisualisasikan matriks korelasi seperti gambar berikut:

```
In [28]: #Check Correlation
f,ax = plt.subplots(figsize=(25, 15))
sns.heatmap(dataset.corr(), annot=True, linewidths=.3, fmt= '.2f',ax=ax, cmap='viridis')
plt.show()
```



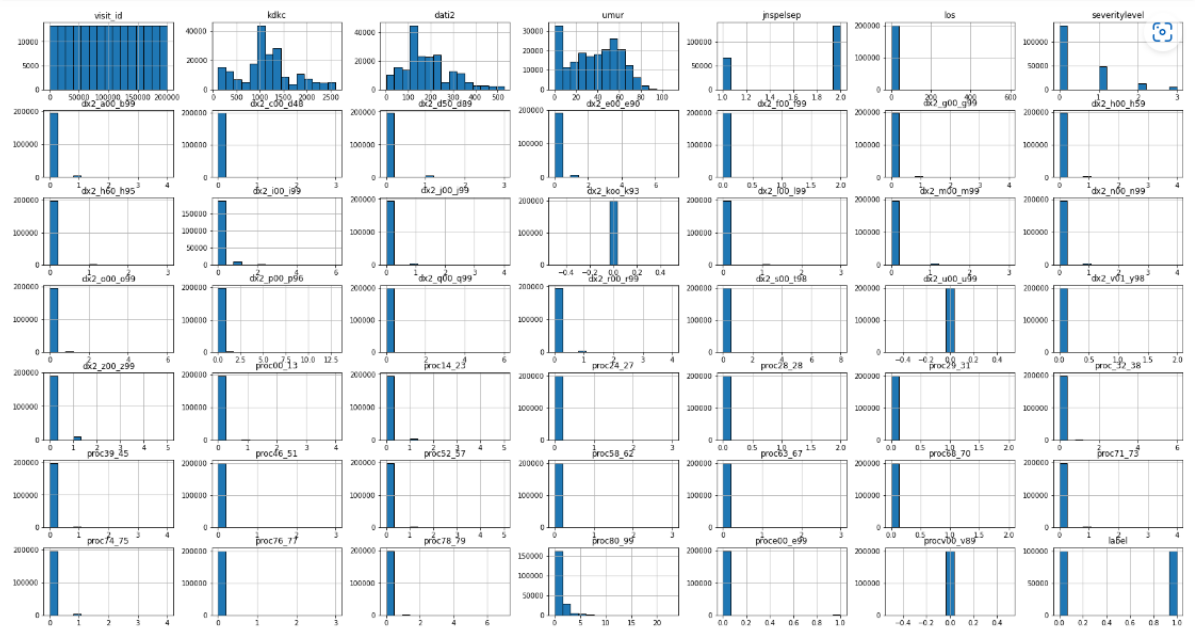
Gambar 2 Correlation

2.3.2. Persebaran Data

Berikut hasil visualisasi penyebaran data pada *histogram* masing-masing label. Penyebaran data cukup beragam untuk semua *dataset*. Untuk melihat persebaran data menggunakan *hist.*

```
# Menampilkan histogram dari semua atribut untuk melihat distribusi setiap atribut
dataset.hist(edgecolor = 'black', bins=15, figsize=(30,16));
```

Gambar 3 Potongan Kode Menampilkan Histogram



Gambar 4 Histogram fitur

2.4 Verify Data Quality

Pada tahapan ini akan dilakukan verifikasi terkait kualitas data yang akan digunakan yaitu data cleaning. Data cleaning merupakan tahapan untuk pembersihan data yang tidak konsisten atau tidak relevan, menghaluskan data yang kasar (noisy), menangani data yang hilang (missing value) atau diberi kode sebagai non-respons, dan mengatasi inkonsistensi data biasanya melibatkan unit pengukuran yang tidak standar atau inkonsistensi nilai. Proses data cleaning dapat menjamin kualitas data dengan mempertimbangkan faktor yang mempengaruhi kualitas data. Hasil penelusuran yang dilakukan pada dataset yang digunakan menemukan jika tidak terdapat atribut yang memiliki nilai null atau kosong.

BAB 3 DATA PREPARATION

Data preparation adalah tahapan untuk memperbaiki masalah yang terdapat pada data sebelum data masuk ke tahap modeling sehingga menghasilkan modeling yang bagus. Pada *data preparation* dilakukan *preprocessing* untuk menjaga kualitas data. *Data preparation* berkaitan dengan semua kegiatan untuk membangun *dataset* seperti *selection, data cleaning, data reduction, integration* yang dapat digunakan dalam model. Berikut adalah hasil yang dapat diperoleh dari Correlation heatmap fitur data klaim pelayanan RS:

- *Feature Selection* : Pada tahap ini ditemukan beberapa data yang hanya memiliki satu nilai data. Fitur ini ('dx2_koo_k93', 'dx2_u00_u99', 'procv00_v89') akan dieliminasi dari data.
- Fitur 'dati2' dan kdkc menjelaskan mengenai wilayah rumah sakit yang melakukan klaim BPJS dan memiliki nilai korelasi yang tinggi yaitu 0.735 sehingga fitur 'dati2' dieliminasi dari data.
- Fitur umur dan *los* (*length of stay*) merupakan nilai numerik dan sebaran data tidak merata. Di tahap ini akan dilakukan transformasi data menjadi bentuk *binning* yaitu data numerik dikelompokkan menjadi beberapa bin agar sebaran data menjadi lebih mudah dipahami. Umur dibagi menjadi 5 kategori berdasarkan pengkategorian usia dari badan WHO (*World Health Organization*). Hasil bin yang diperoleh : Bin 1: umur <=1, Bin 2: 2<= umur >=10, Bin 3: 11<= umur >=19, Bin 4: 20<= umur >=60, Bin 5: umur > 60). *los* dengan fitur 'jnspelsep' (rawat inap atau rawat jalan) diperoleh nilai 0 yang merupakan pasien rawat jalan. Hasil bin yang diperoleh ada 3: 1-5 *short stay*, 6-10 *medium stay*, >10 *long stay*.

3.1 Dataset Description

Pada tahap ini, akan dijalankan beberapa fungsi untuk mendeskripsikan dataset:

1. **.describe()**

   Fungsi ini digunakan untuk menghitung beberapa data statistik seperti:

   - count untuk menghitung jumlah nilai yang tidak kosong.
   - mean yaitu nilai rata-rata
   - std yaitu standar deviasi
   - min merupakan nilai minimum

- 25% merupakan persentil 25%*
- 50% merupakan persentil 50%*
- 75% merupakan persentil 75%*
- maks merupakan nilai maksimum

```
In [14]: # Menampilkan statistik deskriptif
         dataset.describe()
```

Out[14]:

	visit_id	kdkc	dati2	umur	jnspelsep	los	severitylevel	dx2_a00_b99	dx2_c00_d48	dx2_d50_d8
count	200217.000000	200217.000000	200217.000000	200217.000000	200217.000000	200217.000000	200217.000000	200217.000000	200217.000000	200217.00000
mean	100109.000000	1147.387816	184.793309	36.850602	1.669778	1.303356	0.444003	0.024893	0.008341	0.02070
std	57797.813761	574.488224	107.226676	23.095928	0.470294	5.639751	0.725227	0.162484	0.093386	0.14684
min	1.000000	101.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	50055.000000	903.000000	114.000000	18.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.00000
50%	100109.000000	1101.000000	169.000000	39.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.00000
75%	150163.000000	1314.000000	232.000000	56.000000	2.000000	2.000000	1.000000	0.000000	0.000000	0.00000
max	200217.000000	2606.000000	528.000000	109.000000	2.000000	592.000000	3.000000	4.000000	3.000000	3.00000

8 rows × 49 columns

Gambar 5 Fungs .describe()

2. .head()

Fungsi head() digunakan untuk mendapatkan n baris pertama.

```
In [2]: dataset.head()
```

Out[2]:

	visit_id	kdkc	dati2	typeppk	jkpst	umur	jnspelsep	los	cmg	severitylevel	...	proc63_67	proc68_70	proc71_73	proc74_75	proc76_77	proc78_79	proc
0	1	1107	150	SB	P	64	2	0	F	0	...	0	0	0	0	0	0	
1	2	1303	200	C	L	45	1	9	E	3	...	0	0	0	0	0	0	
2	3	1114	172	B	P	34	2	0	Q	0	...	0	0	0	0	0	0	
3	4	601	90	SC	L	34	2	0	Q	0	...	0	0	0	0	0	0	
4	5	1006	130	B	L	27	2	0	F	0	...	0	0	0	0	0	0	

5 rows × 53 columns

Gambar 6 Fungsi .head()

3. .info()

Fungsi ini digunakan untuk menampilkan gambaran *dataset*. Fungsi ini menampilkan informasi tentang DataFrame termasuk index dtype dan column dtypes, non-null values dan memory usage.

```
In [3]: dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200217 entries, 0 to 200216
Data columns (total 53 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   visit_id       200217 non-null  int64
 1   kdkc           200217 non-null  int64
 2   dati2          200217 non-null  int64
 3   typeppk        200217 non-null  object
 4   jkpst          200217 non-null  object
 5   umur           200217 non-null  int64
 6   jnspelsep      200217 non-null  int64
 7   los            200217 non-null  int64
 8   cmg            200217 non-null  object
 9   severitylevel  200217 non-null  int64
 10  diagprimer     200217 non-null  object
 11  dx2_a00_b99    200217 non-null  int64
 12  dx2_c00_d48    200217 non-null  int64
 13  dx2_d50_d89    200217 non-null  int64
 14  dx2_e00_e90    200217 non-null  int64
 15  dx2_f00_f99    200217 non-null  int64
 16  dx2_g00_g99    200217 non-null  int64
 17  dx2_h00_h59    200217 non-null  int64
 18  dx2_h60_h95    200217 non-null  int64
 19  dx2_i00_i99    200217 non-null  int64
 20  dx2_j00_j99    200217 non-null  int64
 21  dx2_koo_k93    200217 non-null  int64
 22  dx2_l00_l99    200217 non-null  int64
 23  dx2_m00_m99    200217 non-null  int64
 24  dx2_n00_n99    200217 non-null  int64
 25  dx2_o00_o99    200217 non-null  int64
 26  dx2_p00_p96    200217 non-null  int64
 27  dx2_q00_q99    200217 non-null  int64
 28  dx2_r00_r99    200217 non-null  int64
 29  dx2_s00_t98    200217 non-null  int64
 30  dx2_u00_u99    200217 non-null  int64
 31  dx2_v01_y98    200217 non-null  int64
 32  dx2_z00_z99    200217 non-null  int64
 33  proc00_13      200217 non-null  int64
 34  proc14_23      200217 non-null  int64
 35  proc24_27      200217 non-null  int64
 36  proc28_28      200217 non-null  int64
 37  proc29_31      200217 non-null  int64
 38  proc_32_38     200217 non-null  int64
 39  proc39_45      200217 non-null  int64
 40  proc46_51      200217 non-null  int64
 41  proc52_57      200217 non-null  int64
 42  proc58_62      200217 non-null  int64
 43  proc63_67      200217 non-null  int64
 44  proc68_70      200217 non-null  int64
 45  proc71_73      200217 non-null  int64
 46  proc74_75      200217 non-null  int64
 47  proc76_77      200217 non-null  int64
 48  proc78_79      200217 non-null  int64
 49  proc80_99      200217 non-null  int64
 50  proce00_e99    200217 non-null  int64
 51  procv00_v89    200217 non-null  int64
 52  label          200217 non-null  int64
dtypes: int64(49), object(4)
memory usage: 81.0+ MB
```

Gambar 7 Fungsi .info()

3.2 Clean Data

Pada tahapan ini dilakukan pembersihan data. Proses *clean data* yaitu proses analisa kualitas dari suatu data dengan cara mengubah, mengoreksi atau menghapus data-data yang tidak sesuai dengan kebutuhan penelitian.

Pada tahapan ini, 3 proses utama yang dilakukan adalah :

- Mengisi nilai-nilai yang hilang

- Mengenali *outliers* dan membersihkan *noisy* data
- Membersihkan redundansi yang disebabkan oleh integrasi data

- Pertama, akan diperiksa apakah di dalam dataset memiliki potensi data yang tidak valid. **isnull()** digunakan untuk memeriksa apakah tidak ada nilai yang tidak valid dalam kumpulan data. *Output* yang dihasilkan yaitu *False* atau *True*.

```
In [6]: dataset.isnull()
```

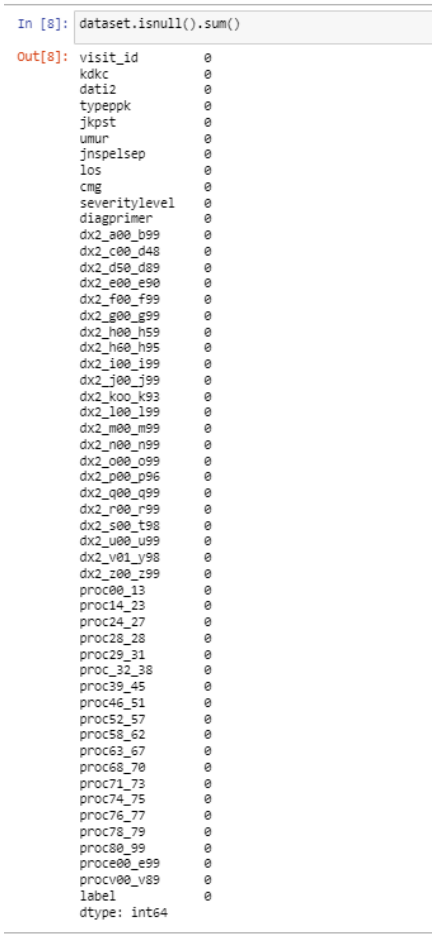
Out[6]:

	visit_id	kdkc	dati2	typeppk	jkpst	umur	jnspelsep	los	cmg	severitylevel	...	proc63_67	proc68_70	proc71_73	proc74_75	proc76_77	proc78
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	Fa
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	Fa
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	Fa
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	Fa
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	Fa
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
200212	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	Fa
200213	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	Fa
200214	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	Fa
200215	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	Fa
200216	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	Fa

200217 rows × 53 columns

Gambar 8 Fungsi .isnull()

- Selanjutnya, untuk menampilkan jumlah *cell* yang memiliki data yang tidak valid, dapat dilakukan dengan fungsi **.sum()**. Dengan menggunakan fungsi .**sum()**, maka akan diketahui berapa jumlah data yang missing value dan berasal dari atribut apa



Gambar 9 Fungsi .isnull().sum()

Code tersebut menjelaskan bahwa tidak terdapat data yang tidak valid atau data *missing value* pada *dataset* yang digunakan, hal ini terlihat *missing value* pada setiap atribut dataset berjumlah 0.

3.3 *Select Data (Data Reduction)*

Pada bagian *data reduction* menjalankan data yang dilakukan dengan cara pengurangan dimensi *dataset* yang tidak sesuai tetapi membentuk hasil yang sama. Maka dapat mengembangkan efisiensi *data mining*, disebabkan *data* yang diproses sedikit. Atribut jkpst dan jnspelsep adalah atribut yang tidak sesuai.

```
col_to_remove = ['visit_id', 'jkpst', 'dx2_koo_k93', 'dx2_u00_u99', 'procv00_v89']
dataset.drop(col_to_remove, axis=1, inplace=True)
```

Gambar 10 Potongan kode untuk menghapus atribut tidak relevan

3.4 *Mapping Attribute*

Data yang dipakai termuat beberapa kolom yang merupakan hasil varian dari tipe atribut. Terkait hal tersebut tersebut dapat diamati dari setiap kolom, Pada kolom diawali dengan penamaan ‘dx’ yang merupakan satu kesatuan dari atribut *secondary* dan juga diawali dengan penamaan ‘proc; yang merupakan dari atribut *procedure.* Proses tersebut bertujuan agar dapat meningkatkan efisiensi data mining disebabkan dataset yang digunakan lebih sedikit.

```
# Melakukan mapping untuk setiap kolom yang diawali 'dx' dan 'proc'
diag_col = dataset.columns[dataset.columns.str.contains(pat = 'dx')].tolist()
proc_col = dataset.columns[dataset.columns.str.contains(pat = 'proc')].tolist()
```

```
def totals(x, cols):
  sum = 0
  for i in cols:
    sum = sum + x[i]
```

```
dataset['total_diagsec'] = dataset[diag_col].sum(axis=1)
dataset['total_proc'] = dataset[proc_col].sum(axis=1)
```

Gambar 11 Potongan kode Mapping Attribute

3.5 *Scaler*

Tahapan pada *Scaler* menggunakan *Minaxcaler* di manfaatkan untuk mengkonversi skala data berkisar antara 0 dan 1. Dataset sebelumnya dibangun menerapkan algoritma *support Vector Machine* untuk melakukan penilaian atau prediksi.

```
from sklearn import preprocessing

scaler_age = preprocessing.MinMaxScaler()
scaler_los = preprocessing.MinMaxScaler()
scaler_diagsec = preprocessing.MinMaxScaler()
scaler_procedure = preprocessing.MinMaxScaler()
```

```
minmax_age = scaler_age.fit(dataset[['umur']])
minmax_los = scaler_los.fit(dataset[['los']])
minmax_diagsec = scaler_diagsec.fit(dataset[['total_diagsec']])
minmax_proc = scaler_procedure.fit(dataset[['total_proc']])
```

```
dataset['umur'] = minmax_age.transform(dataset[['umur']])
dataset['los'] = minmax_los.transform(dataset[['los']])
dataset['total_diagsec'] = minmax_diagsec.transform(dataset[['total_diagsec']])
dataset['total_proc'] = minmax_proc.transform(dataset[['total_proc']])
```

Gambar 12 Potongan kode Scaller

BAB 4 MODELLING

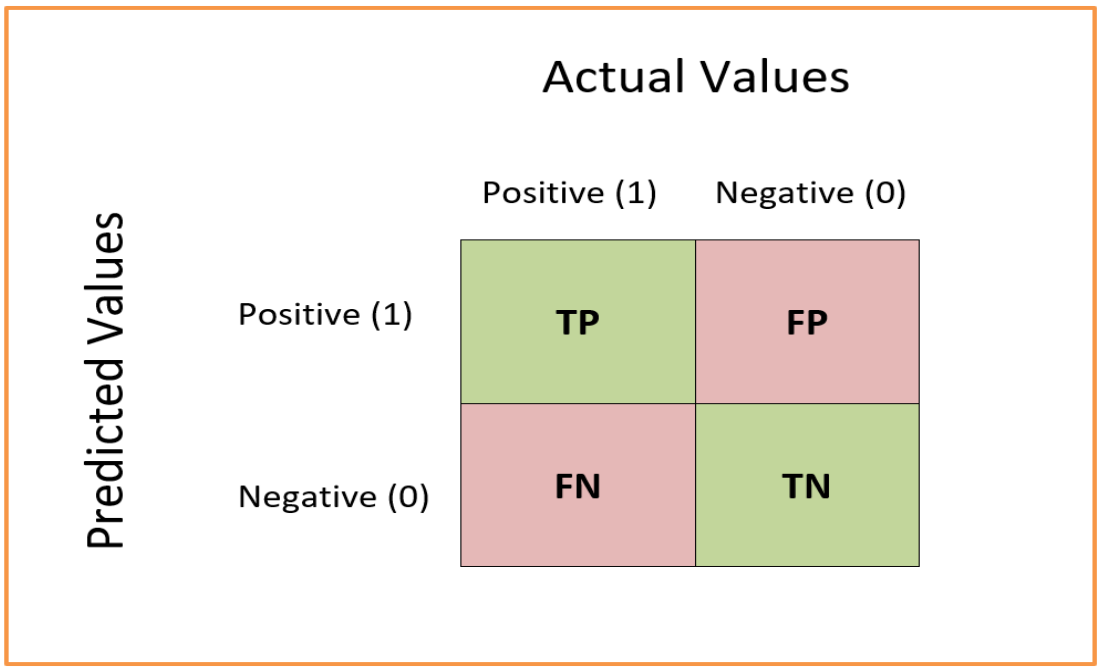
Pada bab sebelumnya, tim telah mempersiapkan data untuk dapat digunakan membangun model. Pada fase ini, teknik pemodelan yang berbeda dipilih dan diterapkan, dan beberapa parameter disesuaikan untuk mendapatkan nilai yang baik. Secara khusus, ada beberapa teknik berbeda yang dapat diterapkan pada masalah *data mining* yang sama.

4.1 Select Modelling Techniques

Pada proyek ini akan menggunakan model klasifikasi *Support Vector Machine* (SVM). *Support Vector Machine* (SVM) adalah sekumpulan metode *supervised learning* yang digunakan untuk *classification, regression* dan *outliers detection*. Teknik pemodelan yang dipakai disesuaikan pada tujuan yang ingin diperoleh dalam melakukan prediksi terkait potensi dan kecurangan pada klaim pelayanan Rumah Sakit. Sebelumnya *dataset* yang ditentukan sudah *balance* dan tidak ada nilai *null* atau *missing value* serta kolom yang tidak diperlukan dihapus.

4.2 Generate Test Design

Setelah melakukan pemilihan model yang akan digunakan, perlu dilakukan pengujian terhadap kualitas model yang akan digunakan. Pengujian model ini dapat dilakukan dengan menggunakan pengukuran yaitu *confusion matrix, accuracy, precision* dan *recall*. *Confusion Matrix* adalah salah satu metode yang digunakan untuk mengukur kinerja dari sistem klasifikasi. Pada *Confusion matrix* akan menyajikan jumlah dari *True Positive (TP), False Positive(FP), True Negative(TN)* dan *False Negative (FN)*. Tabel Confusion Matrix dapat dilihat pada gambar dibawah ini :



Gambar 13 Confusion Matrix

Berikut keterangan dari gambar diatas :

- *True Positive* (TP) merupakan banyaknya data dengan nilai sesungguhnya adalah positif dan nilai prediksi positif.
- *False Positive* (FP) merupakan banyaknya data dengan nilai sesungguhnya adalah negatif dan nilai prediksi positif.
- *True Negative* (TN) merupakan banyaknya data dengan nilai sesungguhnya adalah negatif dan nilai prediksi negatif.
- *False Negative* (FN)merupakan banyaknya data dengan nilai adalah positif dan nilai prediksi negatif.

4.3 Build Model

Pada tahap build model akan dibangun model *Support Vector Machine* dengan menggunakan dataset melalui *preprocessing* dengan menghapus beberapa atribut yang tidak relevan.

Features Selection

Pada *features selection* menggunakan fungsi **.iloc()** yang berfungsi untuk memilih baris atau kolom tertentu dari kumpulan data. X akan mengambil semua data kecuali kolom label. y akan mengambil semua kolom label.

```
# X mengambil semua data kecuali kolom label
X = dataset.iloc[:, dataset.columns != 'label']
# y mengambil kolom label
y = dataset.iloc[:, 9].values
```

Gambar 14 Potongan kode Feature Selection

Splitting Data

Untuk fungsi ini, tim menggunakan library python yaitu **train_test_split** dari **sklearn**. Dimana untuk *test set* sebanyak 20% dari total data dan untuk *train set* sebanyak 80%.

```
# Membagi data menjadi data latih dan data uji
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=47)
```

Berikut *output* hasil splitting data :

```
print("Banyak data latih: ", len(X_train))
print("Banyak data uji: ", len(X_test))
```

```
Banyak data latih:  160173
Banyak data uji:  40044
```

Gambar 15 Potongan Kode Splitting Data

Generating Model

Model akan dibangun dengan menggunakan *library python* yaitu SVC()

```
from sklearn.model_selection import GridSearchCV, StratifiedKFold
from sklearn.svm import SVC
from sklearn import svm
```

```
cv = StratifiedKFold(
    n_splits=10, random_state=0, shuffle=True
)
```

Gambar 16 Potongan kode Support Vector Machine (1)

SVM digenerate dengan menggunakan kernel *Radial Basis Function* (RBF) dengan menggunakan dua parameter *Cost* (C) dan gamma. Parameter *Cost* (C) merupakan parameter yang berfungsi sebagai optimasi SVM untuk menghindari kesalahan klasifikasi pada setiap sampel *dataset training*. Parameter gamma menentukan seberapa jauh pengaruh dari satu sampel *dataset*

pelatihan. Pada implementasi gamma yang digunakan senilai 1000 dan c yang digunakan senilai 1.

```
# C=1
svc=SVC(kernel='rbf',gamma = 1000, C = 1)

# fit classifier to training set
svc.fit(X_train,y_train)
```

```
SVC(C=1, gamma=1000)
```

Gambar 17 Potongan kode Support Vector Machine (2)

4.4 Assess Model

Assess Model merupakan prosedur yang dikerjakan untuk mengevaluasi berdasarkan model yang sudah dibangun sesuai kriteria yang telah dirancang. Kriteria dari prediksi potensi terjadinya *fraud* pada kalin pelayanan rumah sakit diperoleh data sebagai berikut:

- *Predicision > 0.60*
- *Accuracy > 0.60*
- *Recall > 0.65*

Membangun model menggunakan *Support Vector Machine* (SVM) telah memperoleh kriteria yang sudah disesuaikan sebelumnya. Sehingga diharapkan memperoleh nilai *accuracy*, *precision* dan *recall yang* telah melampaui batas kriteria.

BAB 5 EVALUATION

Bab ini akan menjelaskan terkait model yang sudah terbentuk dan diinginkan mempunyai kualitas yang baik untuk analisis data. Prosedur bab ini akan mengevaluasi kinerja dan kualitas model sebelum menetapkan model tersebut sesuai atau tidak dalam mencapai tujuan yang ditetapkan di awal (*Business Understanding)*.

5.1 Evaluate Results

Evaluate Results merupakan tahapan yang menghasilkan model yang telah dibangun menggunakan algoritma Support Vector Machine (SVM). Pada proses ini menggunakan *accuracy_score, precision_score, recall_score*. Pada model dalam mengimplementasi menggunakan bahasa pemrograman python. Model dibangun menggunakan algoritma SVM pada *accuracy_score, precision_score, recall_score* dapat dilihat hasil yang diperoleh, sebagai berikut:

Berikut hasil *accuracy_score, precision_score, recall_score* pada ***training*** :

```
svc_predicts = predict_model(svc, X_train, y_train)
```

```
Confusion Matrix :
[[68726 11073]
 [10267 70107]]

Accuracy Score :  0.8667690559582452

Precision Score :  0.86359940872136

Recall Score :  0.8722596859680991
```

Gambar 18 Output Training

Berikut hasil *accuracy_score, precision_score, recall_score* pada ***testing***:

```
from sklearn import metrics
y_pred = svc.predict(X_test)
print("Test Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Test Accuracy: 0.6447907301967836
```

Gambar 19 Output Testing

Hasil tersebut pada pembangunan model diperkirakan berdasarkan kesesuaian parameter yang telah ditetapkan termasuk *minimum accuracy*, *minimum precision* dan *minimum recall*. Model yang dibangun dalam mengimplementasikan algoritma *Support Vector Machine* untuk

memprediksi *fraud* masih mengalami *overfitting* dimana proses *training* memperoleh kinerja baik sedangkan testing sebaliknya.

5.2 Evaluate Process

Evaluate Process merupakan tahapan memvalidasi dari awal untuk memastikan agar tidak ada faktor yang terlewatkan. Berdasarkan hasil yang diperoleh dari proses awal proyek *data mining* menggunakan metode CRISP-DM, maka dapat dipahami bahwa:

- Proses eksplorasi data dengan mudah untuk menentukan atribut yang berhubungan dalam memprediksi potensi terjadinya *fraud* pada klaim pelayanan Rumah Sakit.
- *Data Preparation*, terutama pada proses *data cleaning* dan *data reduction.* Maka data yang didapat menciptakan model yang baik.
- Pentingnya agar tetap fokus terhadap masalah bisnis yang dibahas, sebab data yang sudah selesai dianalisis selanjutnya adalah tahap *modelling.* Dapat kita pahami bahwa *Business understanding* begitu penting untuk menentukan hasil yang dibutuhkan untuk memprediksi potensi terjadinya *fraud* pada klaim pelayanan Rumah Sakit.

BAB 6 DEPLOYMENT

Pada tahap ini, knowledge atau informasi yang telah diperoleh akan diatur dan dipresentasikan dalam bentuk khusus sehingga dapat digunakan oleh user. Tahap deployment dapat berupa pembuatan laporan sederhana atau mengimplementasikan proses data mining yang berulang.

DAFTAR PUSTAKA

1. Heppy Maria Simanungkalit, dkk,"Deteksi FraudPada Klaim Layanan Rumah SakitMenggunakan Model Neural Network",vol.1, no.1, 2021