LAPORAN PROYEK DATA MINING

Binary Classification using Random Forest

Disusun Oleh:

12S18018 Yohana Polin Simatupang

12S18019 Maria Puspita Sari Nababan

12S18064 Letare Aiglien Saragih

PROGRAM STUDI SARJANA SISTEM INFORMASI

FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO

INSTITUT TEKNOLOGI DEL

2021

BAB 1
BUSINESS UNDERSTANDING

Pada pengerjaan proyek ini akan dilakukan sesuai dengan tahapan pada metodologi CRISP DM yang akan dimulai dengan tahapan *business understanding* yaitu memahami permasalahan bisnis untuk proses *data mining* yang akan dilakukan. Adapun yang termasuk bagian dari tahapan ini adalah menentukan tujuan bisnis, menentukan sasaran yang ingin dicapai dengan data mining, dan menghasilkan perencanaan proyek yang akan dilakukan.

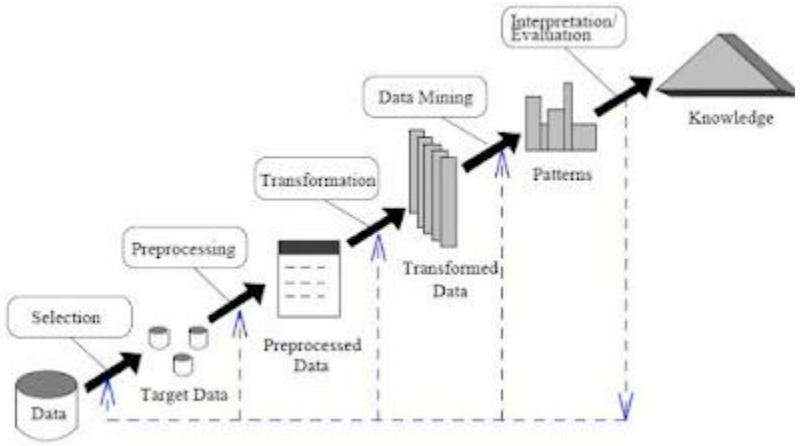
1.1 Determine Business Objective

Rumah sakit merupakan salah satu instansi yang bergerak sebagai pelayanan kesehatan bagi masyarakat. Dalam melaksanakan proses bisnisnya, peran BPJS cukup besar dalam mempengaruhi kualitas pelayanan bagi masyarakat. Namun dengan semakin banyak penggunaan BPJS Kesehatan, tidak jarang terjadi beberapa kecurangan (*fraud*) yang ditujukan untuk menguntungkan pihak tertentu. Pelaku yang terlibat bisa jadi adalah peserta BPJS Kesehatan, *fasilitator* kesehatan atau pembeli layanan kesehatan, penyedia obat dan alat kesehatan, dan pemangku kepentingan lainnya. Penanganan terkait masalah tersebut menjadi *concern* yang perlu untuk diatasi yang bertujuan untuk dapat mencegah dan mendeteksi berbagai indikasi potensi kecurangan sedini dan sesedikit mungkin. Sehingga dengan demikian biaya pelayanan kesehatan dapat dimanfaatkan semaksimal mungkin dalam memenuhi kepentingan dan pelayanan yang maksimal bagi masyarakat, serta untuk tetap menjaga *sustainability* BPJS Kesehatan,

1.2 Determine Data Mining Goal

Tujuan bisnis pada penelitian ini adalah untuk melakukan prediksi potensi terjadinya penyimpangan (*fraud*) pada klaim pelayanan Rumah Sakit. Melihat jumlah data yang besar dan studi kasus yang akan diteliti untuk itu, dilakukan penerapan *data mining* untuk menemukan pola menarik dari data. *Data mining* dikelompokkan menjadi *description, estimation, prediction, classification, clustering*, dan *association* [ref: buku pang-ning tan]. Pada penelitian ini, penggunaan data mining bertujuan sebagai dasar dalam pengembangan sebuah model klasifikasi biner untuk menemukan fraud. Ketika melakukan proses data mining, harus dilakukan beberapa tahapan antara lain, pembersihan data, integrasi data,

pemilihan data, transformasi data, penemuan pola, evaluasi pola dan presentasi pengetahuan.



Tahapan Knowledge Discovery in Databases (KDD)

Dalam menemukan faktor apa saja yang menyebabkan terjadinya penyimpangan (*fraud*) pada layanan BPJS perlu digunakan data mining task dengan teknik asosiasi. *Association rule mining* adalah metode pembelajaran mesin berbasis aturan untuk menemukan hubungan yang menarik antara variabel dalam data yang berjumlah besar.

Algoritma yang akan untuk penelitian ini adalah Algoritma *Random Forest Classifier* (RFC). RFC merupakan metode klasifikasi yang *supervised* menggabungkan ratusan atau ribuan pohon keputusan, melatih masing-masing pohon pada serangkaian pengamatan yang sedikit berbeda, memisahkan simpul di setiap pohon dengan mempertimbangkan sejumlah fitur yang terbatas. Prediksi akhir dari random forest dibuat dengan merata-ratakan prediksi dari masing-masing pohon. Kelebihan dari algoritma ini adalah: menghasilkan eror yang lebih rendah, memberikan hasil yang bagus dalam klasifikasi, dapat mengatasi data training dalam jumlah sangat besar secara efisien, efektif untuk mengestimasi hilangnya data, memperkiraan variabel apa yang penting dalam klasifikasi dan menyediakan metode eksperimental untuk mendeteksi interaksi variabel.

1.3 Produce Project Plan

Tahap perencanaan yang dilakukan untuk mencapai tujuan data mining dan mencapai tujuan bisnis pada penelitian '*Binary Classification using Random Forest*' ini adalah sebagai berikut:

Aktivitas	Sub Aktivitas	Durasi	Sumber daya yang dibutuhkan	Ketergantungan
Pemilihan Kasus dan Algoritma	Pemilihan Kasus	1	Semua analisis	-
	Penentuan Algoritma	6		
Business Understanding	Menentukan Objektif Bisnis	1	Semua analisis	Pemilihan kasus dan algoritma
	Menentukan Tujuan Bisnis	1		
	Membuat Rencana Proyek	1		
Data Understanding	Mengumpulkan Data	1	Semua analisis	Data dan teknologi
	Menelaah Data	1		
	Memvalidasi Data	1		
Data Preparation	Memilah Data	1	*Data mining consultant,* beberapa *database analyst time*	Data dan teknologi
	mengkonstruksi Data	4		
	Menentukan Label Data	1		
	Membersihkan Data	4		
Modeling	Membangun Skenario Pengujian	3	*Data mining consultant,* beberapa *database analyst time*	Algoritma
	Membangun Model	7		
Model Evaluation	Mengevaluasi Hasil Pemodelan	5	Semua analisis	Model yang telah dibuat

	Melakukan Review Proses Pemodelan	4		
Deployment	melakukan Deployment Model	2	*Data mining consultant,* beberapa *database analyst time*	Penerapan model berdasarkan data dan algoritma yang dipilih
	Membuat laporan akhir Proyek	4		

Dalam pelaksanaan proyek dalam penelitian ini, diperlukan tools data mining yang mendukung metode untuk berbagai tahapan proses. Tools dan teknik yang digunakan dapat mempengaruhi keseluruhan proyek. Tools yang digunakan dalam mengerjakan proyek ini adalah python. Python adalah bahasa pemrograman berorientasi objek yang digunakan dalam pengembangan perangkat lunak maupun dalam analisis dan data science. Python memiliki berbagai library yang menyediakan fungsi untuk melakukan analisis data, memproses data, memvisualisasikan data, dll.

BAB 2
DATA UNDERSTANDING

Tahap kedua pada metodologi CRISP-DM setelah *business understanding* dalam melakukan metodologi *data science* adalah *data understanding*. Pada bab ini akan dijelaskan mengenai pengumpulan *initial data*, analisis untuk dapat memahami data yang akan digunakan dalam penelitian serta verifikasi pada kualitas data.

2.1 Collect Initial Data

Langkah *data understanding* diawali dengan pengumpulan data yang akan digunakan pada proses *data science.* Data yang akan digunakan dalam kasus *binary classification* menggunakan *Random Forest Classification* (RFC) adalah data BPJS Kesehatan yang berasal dari dataset yang digunakan dalam kompetisi Hackathon.

2.2 Analysis Data

Dataset train yang digunakan untuk memprediksi penyimpangan (fraud) pada layanan BPJS terdiri dari 200217 observasi dan 53 variable. Kemudian perlu dilakukan Exploratory Data Analysis (EDA). EDA digunakan untuk memahami data, mendapatkan konteks data, memahami variabel dan hubungan di antara variabel, dan merumuskan hipotesis yang berguna dalam membangun model prediksi. Atribut atau fitur pada dataset tidak semua diperlukan dalam menganalisis. Fitur atau atribut yang digunakan merupakan atribut yang relevan dan sesuai dengan tujuan proyek. Adapun fitur atau variabel yang dibutuhkan dalam penelitian ini antara lain, yaitu:

No	Variabel	Tipe Variabel	Deskripsi
1	visit_id		id kunjungan
2	kdkc		kode wilayah kantor cabang BPJS Kesehatan
3	dati2		kode kabupaten/kota
4	typeppk		kode tipe Rumah Sakit

5	jkpst		jenis kelamin peserta JKN-KIS
6	umur		umur peserta saat mendapatkan pelayanan rumah sakit
7	jnspelsep		tingkat pelayanan; 1:rawat inap; 2. rawat jalan
8	los		lama peserta dirawat di rumah sakit
9	cmg		klasifikasi CMG (Case Mix Group)
10	severitylevel		tingkat urgensi
11	diagprimer		diagnosa primer
12	dx2_..._...		diagnosa sekunder
13	proc.._...		kode kelompok procedure
14	label		flag fraud; 1:fraud; 0:tidak fraud

Untuk mendapatkan hasil analisa dataset yang lebih baik, makan perlu dilakukan pengidentifikasian kembali subset data yang relevan untuk kemudian digunakan pada tahapan selanjutnya yang sesuai dengan tujuan data mining pada penelitian ini.

2.3 Verify Data Quality

Tahapan selanjutnya adalah melakukan verifikasi terhadap kualitas data yang digunakan. Untuk mendapatkan data yang berkualitas baik, perlu dilakukan pembersihan data (*dat cleaning*). *Data cleaning* pada proses data mining dapat mengurangi jumlah dan kompleksitas data. Salah satu aspek yang menyebabkan kualitas data menjadi kurang baik adalah terjadinya *missing value* atau terdapat data yang hilang pada dataset yang digunakan. Untuk mengantisipasi hal tersebut terlebih dahulu dilakukan pemeriksaan apakah terdapat data yang hilang (*missing*) atau bernilai kosong. Pemeriksaan dilakukan menggunakan fungsi pada python yaitu *df.isna().* Adapun hasil yang didapatkan dari pemeriksaan tersebut adalah bahwa pada dataset yang digunakan tidak terdapat *missing value*.

BAB 3

DATA PREPARATION

3.1 Sorting Data

Data yang akan digunakan dalam proses *data mining* terlebih dahulu perlu dipersiapkan dengan baik. Fase *sorting* merupakan tahapan untuk melakukan pemilihan pada atribut yang akan digunakan. Atribut yang tidak digunakan akan *di drop*.

```
[18] df.drop(['visit_id', 'procv00_v89', 'dx2_koo_k93', 'dx2_u00_u99', 'dati2'], axis=1, inplace=True)
```

Atribut tersebut di *drop* dengan tujuan agar data yang digunakan lebih efisien dan efektif dalam pengolahan data termasuk dalam penggunaan memory. Berikut adalah tampilan setelah atribut yang tidak digunakan telah di *drop*

```
[19] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200217 entries, 0 to 200216
Data columns (total 48 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   kdkc           200217 non-null  int64
 1   typeppk        200217 non-null  object
 2   jkpst          200217 non-null  object
 3   umur           200217 non-null  int64
 4   jnspelsep      200217 non-null  int64
 5   los            200217 non-null  int64
 6   cmg            200217 non-null  object
 7   severitylevel  200217 non-null  int64
 8   diagprimer     200217 non-null  object
 9   dx2_a00_b99    200217 non-null  int64
 10  dx2_c00_d48    200217 non-null  int64
 11  dx2_d50_d89    200217 non-null  int64
 12  dx2_e00_e90    200217 non-null  int64
 13  dx2_f00_f99    200217 non-null  int64
 14  dx2_g00_g99    200217 non-null  int64
 15  dx2_h00_h59    200217 non-null  int64
 16  dx2_h60_h95    200217 non-null  int64
 17  dx2_i00_i99    200217 non-null  int64
 18  dx2_j00_j99    200217 non-null  int64
 19  dx2_l00_l99    200217 non-null  int64
 20  dx2_m00_m99    200217 non-null  int64
 21  dx2_n00_n99    200217 non-null  int64
 22  dx2_o00_o99    200217 non-null  int64
 23  dx2_p00_p96    200217 non-null  int64
 24  dx2_q00_q99    200217 non-null  int64
 25  dx2_r00_r99    200217 non-null  int64
 26  dx2_s00_t98    200217 non-null  int64
 27  dx2_v01_y98    200217 non-null  int64
 28  dx2_z00_z99    200217 non-null  int64
 29  proc00_13      200217 non-null  int64
 30  proc14_23      200217 non-null  int64
 31  proc24_27      200217 non-null  int64
 32  proc28_28      200217 non-null  int64
 33  proc29_31      200217 non-null  int64
 34  proc_32_38     200217 non-null  int64
 35  proc39_45      200217 non-null  int64
 36  proc46_51      200217 non-null  int64
 37  proc52_57      200217 non-null  int64
 38  proc58_62      200217 non-null  int64
 39  proc63_67      200217 non-null  int64
 40  proc68_70      200217 non-null  int64
 41  proc71_73      200217 non-null  int64
 42  proc74_75      200217 non-null  int64
 43  proc76_77      200217 non-null  int64
 44  proc78_79      200217 non-null  int64
 45  proc80_99      200217 non-null  int64
 46  proce00_e99    200217 non-null  int64
 47  label          200217 non-null  int64
dtypes: int64(44), object(4)
memory usage: 73.3+ MB
```

Nilai penggunaan *memory* menjadi berkurang setelah dilakukan pemilihan atribut yang diperlukan yaitu sebagai berikut

```
dtypes: int64(49), object(4)
memory usage: 81.0+ MB
```

```
dtypes: int64(44), object(4)
memory usage: 73.3+ MB
```

3.2 Cleaning Data

Fase ini merupakan tahapan untuk melakukan pembersihan data. Pembersihan data yang dilakukan adalah menangani objek data yang kosong (*missing value*). Untuk itu, terlebih dahulu dilakukan pemeriksaan data dengan menggunakan fungsi `df.isna()`

```
[10] # check missing value
     df.isna()
```

	visit_id	kdkc	dati2	typeppk	jkpst	umur	jnspelsep	los	cmg	severitylevel	diagprimer	dx2_a00_b99	dx2_c00_d48	dx2_d50_d89	dx2_e00_e90	dx2_f0
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
200212	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
200213	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
200214	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
200215	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	
200216	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	

200217 rows × 53 columns

Python Pandas memungkinkan kita dapat menemukan *missing value* secara cepat dengan fungsi `isna()`. Fungsi `isna()` akan mengembalikan nilai boolean dari dataset yang diperiksa. Hasil keluaran berupa **False** menunjukkan bahwa pada cell tersebut tidak terdapat nilai yang kosong (*missing*). Agregasi data dengan fungsi `sum()` ditujukan agar dapat memahami data dengan lebih baik. Agregasi `sum()` akan menjumlahkan semua cell yang kosong apabila terdapat nilai yang kosong pada atribut tertentu.

```
[12] # checking missing value
     df.isna().sum()

     visit_id         0
     kdkc             0
     dati2            0
     typeppk          0
     jkpst            0
     umur             0
     jnspelsep        0
     los              0
     cmg              0
     severitylevel    0
     diagprimer       0
     dx2_a00_b99      0
     dx2_c00_d48      0
     dx2_d50_d89      0
     dx2_e00_e90      0
     dx2_f00_f99      0
     dx2_g00_g99      0
     dx2_h00_h59      0
     dx2_h60_h95      0
     dx2_i00_i99      0
     dx2_j00_j99      0
     dx2_koo_k93      0
     dx2_l00_l99      0
```

3.3 Construct Data

Fase ini merupakan tahapan untuk melakukan konstruksi pada data. Adapun konstruksi yang dilakukan adalah transformasi atribut dengan tipe kategorik menjadi numerik. Hal ini bertujuan agar data kemudian dapat di normalisasi. Untuk tahap pada konstruksi data dilakukan pengecekan tipe data pada dataset menggunakan fungsi `df.info()`, dan output yang dihasilkan adalah sebagai berikut:

```
 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
```

Dapat dilihat pada gambar di atas, terdapat 4 atribut yang bertipe data kategorikal (object64), untuk itu perlu dilakukan transformasi data. Untuk itu perlu dilakukan transformasi data tipe pada atribut dengan menjalankan potongan kode berikut:

```
[12] from numpy.core.defchararray import add
     # bpjs_data with numeric data type
     data_num = df.select_dtypes(include=[np.number])

     # bpjs_data with category data type
     data_cat = df.select_dtypes(exclude=[np.number])

     # Get dummies (data transformation)
     transform_cat = pd.get_dummies(data_cat, prefix_sep='_', drop_first=True)
```

```
[13] data_cat = transform_cat.assign(new=add('', np.arange(1, len(data_cat) + 1).astype(str)))
     data_num = data_num.assign(new=add('', np.arange(1, len(data_num) + 1).astype(str)))
     bpjs_data_final = pd.concat([data_cat, data_num], axis=1)
     bpjs_data_final.drop(['new'], axis=1, inplace=True)
```

Setelah transformasi berhasil, dilakukan pengecekan kembali pada type atribut menggunakan fungsi `df.info()`

```
[14] bpjs_data_final.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200217 entries, 0 to 200216
Columns: 115 entries, typeppk_B  to label
dtypes: int64(49), uint8(66)
memory usage: 87.5 MB
```

BAB 4

MODELLING

4.1 Build Test Scenario

4.2 Model Building

BAB 5

MODEL EVALUATION

5.1 Evaluation of Modeling Result

5.2 Modeling Process Review

BAB 6

DEPLOYMENT

6.1 Model Development

6.2 Final Report