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

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

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.

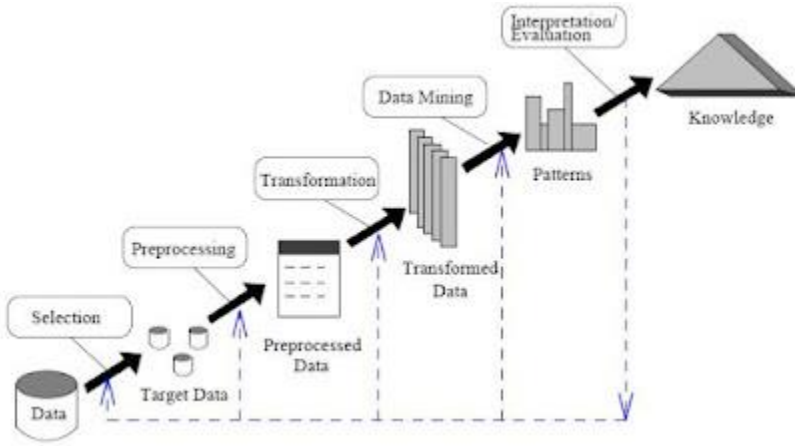


Figure 1 Tahapan *Knowledge Discovery in Databases (KDD)*

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:

Table 1 Tahap perencanaan yang dilakukan untuk mencapai tujuan *data mining* dan mencapai tujuan bisnis pada penelitian '*Binary Classification using Random Forest*'

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. Data yang digunakan dalam penelitian ini merupakan data dengan format csv yang sudah terstruktur. Memuat informasi BPJS Kesehatan yang merupakan data publik mengenai aturan penamaan dan kesehatan secara umum. Data yang digunakan berukuran 10611501

```
df.size
```

```
10611501
```

Figure 2 Memuat informasi ukuran data

2.2 Analysis Data

Dataset train yang digunakan untuk memprediksi penyimpangan (fraud) pada layanan BPJS terdiri dari 200217 observasi dan 53 variabel dan memiliki proporsi kelas label pada data seimbang.

```
# melihat ukuran data
df.shape
```

```
(200217, 53)
```

Figure 3 Memuat informasi bentuk data

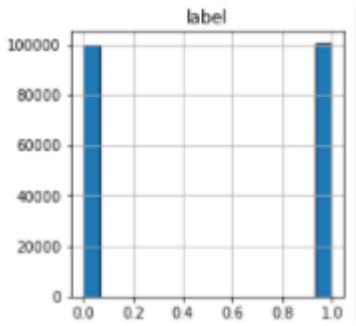


Figure 4 Proporsi kelas Label menggunakan histogram

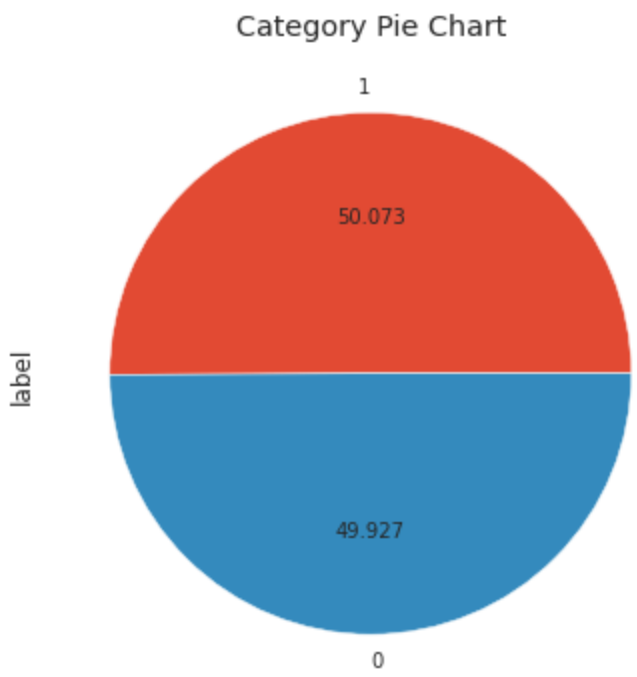
Adapun ke 53 fitur/ variabel yang dimaksud adalah sebagai berikut:

```
[ ]  df.columns

     Index(['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_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', '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',
            'label'],
           dtype='object')
```

Figure 5 Informasi mengenai 53 fitur pada dataset

Pada tahap ini akan dilakukan pendefinisian label data yang akan digunakan. Tahap ini bertujuan untuk memastikan bahwa data yang digunakan merupakan *balance dataset*.




```
1    100255
0     99962
Name: label, dtype: int64
```

Figure 6 Proporsi kelas Label menggunakan Pie Chart

Berdasarkan pie chart diatas, dapat dilihat bahwa jumlah data yang menunjukkan *fraud* dan tidak *fraud* pada feature Label adalah seimbang.

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. Langkah awal yg dilakukan untuk memahami data adalah dengan menganalisis tipe dari setiap fitur/ variabel yang akan digunakan menggunakan fungsi info().

```
# melihat type atribut
df.info()
```

Figure 7 Potongan code untuk melihat tipe atribut

Berdasarkan fungsi tersebut diperoleh informasi mengenai tipe atribut atau fitur pada dataset sebagai berikut:

Table 2 Informasi mengenai atribut, tipe atribut dan keterangan atribut

No	Variabel	Tipe Variabel	Deskripsi
1	visit_id	int64	id kunjungan
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	dx2_..._...	int64	diagnosa sekunder
13	proc.._...	int64	kode kelompok procedure
14	label	int64	flag fraud; 1:fraud; 0:tidak fraud

Dari 53 fitur yang tersedia, terdapat 2 kategori yang diperoleh, yaitu: 4 fitur dengan data kategorik dan 49 fitur dengan data numerik. Untuk mendapatkan hasil analisa dataset yang lebih baik, maka 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*). Sebelum pembersihan data dilakukan, terlebih dahulu dilakukan pengecekan struktur data.

```
[ ] df.head()
```

	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_f00_f99	dx2_g00_g99	dx2_h00_h5
0	1	1107	150	SB	P	64	2	0	F	0	f00_f99	0	0	0	0	0	0	
1	2	1303	200	C	L	45	1	9	E	3	e00_e90	1	0	0	0	0	0	
2	3	1114	172	B	P	34	2	0	Q	0	r00_r99	0	0	0	0	0	0	
3	4	601	90	SC	L	34	2	0	Q	0	r00_r99	0	0	0	0	0	0	
4	5	1006	130	B	L	27	2	0	F	0	f00_f99	0	0	0	0	0	0	

Figure 8 Struktur dataset

untuk mengetahui nilai unik di sepanjang sumbu (*axis*) kolom kita akan menggunakan fungsi nunique() yang akan memprint total nilai unik di setiap baris. Hal ini bertujuan untuk melihat kualitas nilai setiap fitur berdasarkan jumlah nilai pada tiap atribut.

```
[ ] visit_id          200217
    kdkc                 126
    dati2                486
    typeppk               25
    jkpst                  2
    umur                 105
    jnspelsep              2
    los                  142
    cmg                   23
    severitylevel          4
    diagprimer            21
    dx2_a00_b99            5
    dx2_c00_d48            4
    dx2_d50_d89            4
    dx2_e00_e90            7
    dx2_f00_f99            3
    dx2_g00_g99            5
    dx2_h00_h59            5
    dx2_h60_h95            4
    dx2_i00_i99            7
    dx2_j00_j99            5
    dx2_koo_k93            1
    dx2_l00_l99            4
    dx2_m00_m99            4
    dx2_n00_n99            5
    dx2_o00_o99            7
    dx2_p00_p96           14
    dx2_q00_q99            7
    dx2_r00_r99            5
    dx2_s00_t98            8
    dx2_u00_u99            1
    dx2_v01_y98            3
    dx2_z00_z99            6
    proc00_13              5
    proc14_23              6
    proc24_27              4
    proc28_28              3
    proc29_31              3
    proc_32_38             6
    proc39_45              5
    proc46_51              4
    proc52_57              6
    proc58_62              4
    proc63_67              4
    proc68_70              3
    proc71_73              5
    proc74_75              5
    proc76_77              4
    proc78_79              7
    proc80_99             22
    proce00_e99            2
    procv00_v89            1
    label                  2
    dtype: int64
```

Berdasarkan pengecekan diatas, dapat dilihat terdapat 3 atribut yang memiliki jumlah nilai = 1, yaitu: dx2_koo_k93, dx2_u00_u99 dan procv00_v89.

Selanjutnya, dilakukan pengecekan kebersihan data dari kasus seperti *noisy*, *missing value*, dan masalah lainnya. *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()*

```
[ ]  #Data Preprocessing
     #Lakukanla imputasi data dengan nilai mean jika terdapat nilai null (jika tidak ada
     #null, tunjukkan pada program anda).
     C = (df.dtypes == 'object')
     CategoricalVariables = list(C[C].index)

     Integer = (df.dtypes == 'int64')
     Float = (df.dtypes == 'float64')
     NumericVariables = list(Integer[Integer].index) + list(Float[Float].index)

     Missing_Percentage = (df.isnull().sum()).sum()/np.product(df.shape)*100
     print("The number of missing entries before cleaning: " + str(round(Missing_Percentage,5)) + " %")

     The number of missing entries before cleaning: 0.0 %
```

Figure 9 Potongan kode untuk melakukan imputasi data null dengan nilai mean

Adapun hasil yang didapatkan dari pemeriksaan tersebut adalah bahwa pada dataset yang digunakan tidak terdapat *missing value*.

Proses verifikasi kualitas data dilanjutkan dengan menggunakan visualisasi data dengan memanfaatkan fungsi hist untuk menampilkan histogram untuk semua atribut. Histogram dalam tampilan bentuk grafis akan menunjukkan distribusi data secara visual atau seberapa sering suatu nilai yang berbeda itu terjadi dalam suatu kumpulan data (*dataframe*). Histogram menunjukkan distribusi data dengan memplot frekuensi kejadian dalam suatu rentang.

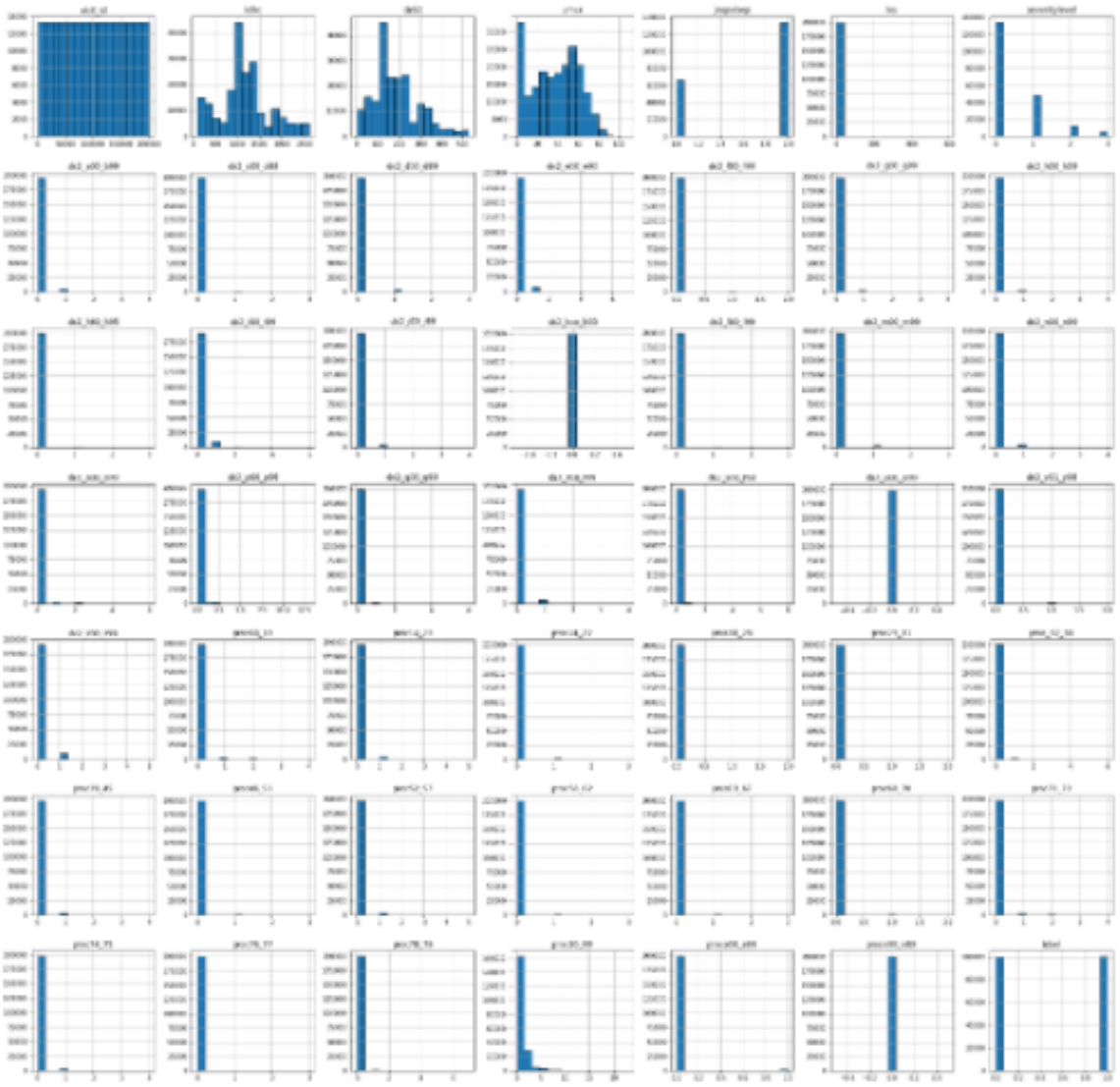


Figure 10 Tampilan histogram untuk setiap fitur pada dataset

Berdasarkan histogram yang dihasilkan, dapat dilihat bahwa kolom yang memiliki variasi data pada frekuensi tertentu adalah kdkc, dati2, dan umur. Namun persebaran atau distribusi data tidak tersebar secara konsisten. Kolom **kdkc** yang menunjukkan kode kantor cabang BPJS yang paling banyak adalah pada code di rentang 1000. Sementara pada kolom **dati2** yang menunjukkan kode kabupaten, paling tinggi berada pada rentang 100-200. Dan untuk kolom **umur**, nilai yang paling tinggi berada pada rentang umur 0.

2.4 Data Correlation

Pada tahap ini, akan dilakukan pengecekan keterkaitan setiap fitur pada data yang digunakan untuk mengetahui bagaimana data akan dimanfaatkan untuk mengatasi masalah bisnis yang akan diselesaikan. Pada fitur severity level dimuat informasi mengenai tingkat urgensi rawat pasien yang dibagi menjadi 4 nilai yaitu 0-3. Urgensi kasus dalam INA-CBG terbagi menjadi:

1. "0" Untuk Rawat jalan
2. "I - Ringan" untuk rawat inap dengan tingkat keparahan 1 (tanpa komplikasi maupun komorbiditi)
3. "II - Sedang" Untuk rawat inap dengan tingkat keparahan 2 (dengan mild komplikasi dan komorbiditi)
4. "III - Berat" Untuk rawat inap dengan tingkat keparahan 3 (dengan major komplikasi dan komorbiditi

Berikut merupakan pie chart yang menampilkan perbandingan dari keempat nilai urgensi tersebut:

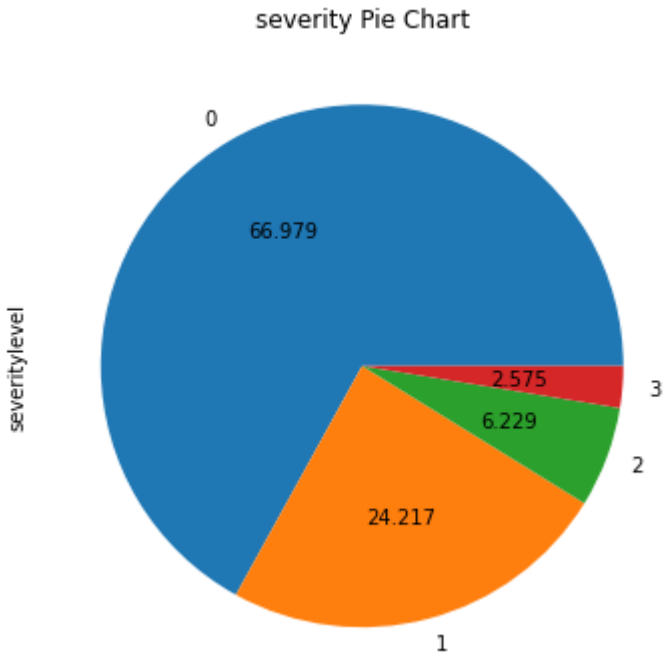


Figure 11 Proporsi kelas severity level

Berdasarkan output tersebut dapat diketahui bahwa nilai paling tinggi ditunjukkan oleh kelas 0 yaitu "rawat jalan". Sub-group tersebut merupakan resource intensity level yang menunjukkan tingkat keparahan kasus yang dipengaruhi adanya komorbiditas ataupun komplikasi dalam masa perawatan.

Kedua adalah melihat korelasi antara fitur los dan umur. Hal ini bertujuan untuk melihat umur pasien yang lebih banyak dirawat.

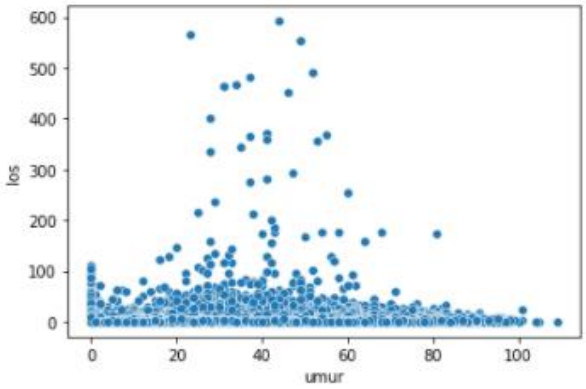


Figure 12 Korelasi antara fitur los dan umur

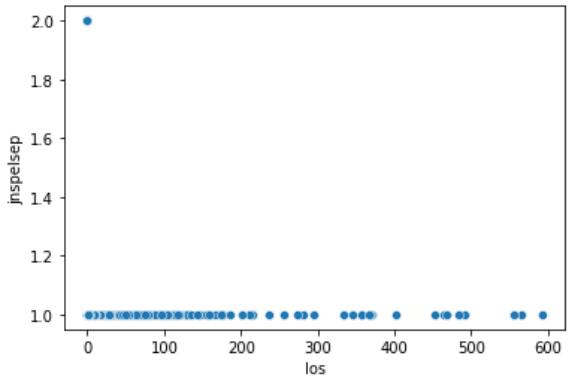


Figure 13 Korelasi antara fitur los dan jnspelsep

Pada atribut jnspelsep terdapat 2 jenis nilai yang digunakan, yaitu 1 dan 2. Nilai 1 diartikan sebagai pasien yang mendapat layanan rawat inap dan nilai 2 diartikan sebagai pasien yang mendapat layanan rawat jalan.

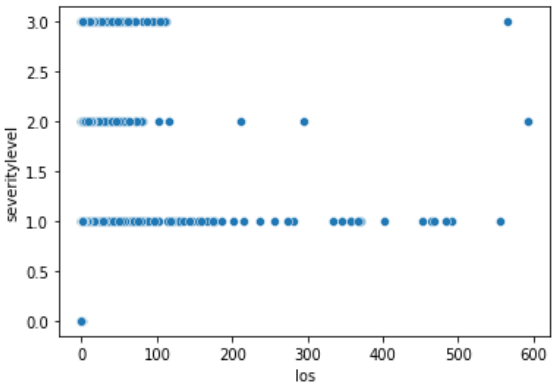


Figure 14 Korelasi antara fitur los dan sevetitylevel

Setelah melakukan proses binning terhadap length of stay (los) yang dibagi menjadi rawat jalan, short stay, medium stay, dan long stay. Dapat dilihat terdapat korelasi tidak valid, saat dirawat jalan maka harusnya rawat jalan hanya ada pada koordinat 2, namun dari hasil visualisasi terdapat tingkat pelayanan rawat jalan yang los nya lebih dari 0 hari, atau menginap.

Keterkaitan (korelasi) tersebut dapat dilihat dengan memvisualisasikan data menggunakan heatmap ataupun scatter plot.

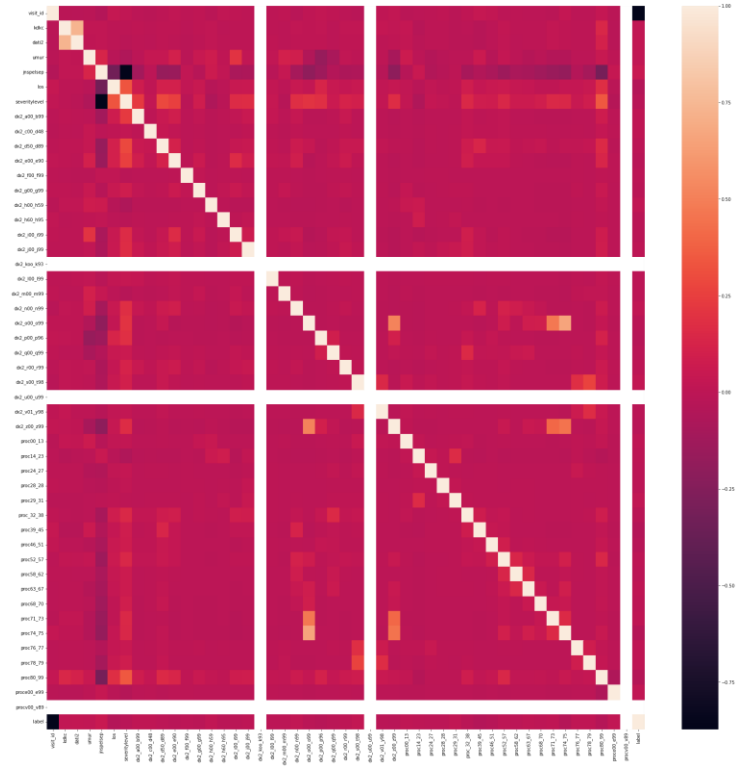


Figure 15 Korelasi setiap fitur pada dataset dengan heatmap

Pada gambar dapat dilihat bahwa 3 atribut dengan jumlah nilai =1 seperti yang disebutkan pada tahap verifikasi data, yaitu: dx2_koo_k93, dx2_u00_u99 dan procv00_v89, tidak memiliki korelasi dengan fitur lain. Sementara 50 fitur lainnya berkorelasi satu sama lain dengan tingkat ketergantungan yang berbeda.

BAB 3
DATA PREPARATION

Tahap ketiga pada metodologi CRISP-DM setelah *data understanding* dalam melakukan metodologi *data science* adalah *data preparation*. Pada bab ini akan dijelaskan mengenai proses apa saja yang akan dilakukan untuk mempersiapkan data seperti *sorting, cleaning, construction, binning* dan *normalization.*

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)
```

Figure 16 Potongan code untuk menghapus fitur tertentu

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()
```

Figure 17 Potongan kode untuk melihat informasi atribut

```
<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
```

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
```

```
[ ]  df.columns
```

```
Index(['kdkc', 'typeppk', 'jkpst', 'umur', 'jnspelsep', 'los', 'cmg',
       'severitylevel', 'diagprimer', '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_l00_l99', 'dx2_m00_m99', 'dx2_n00_n99', 'dx2_o00_o99',
       'dx2_p00_p96', 'dx2_q00_q99', 'dx2_r00_r99', 'dx2_s00_t98',
       'dx2_v01_y98', 'dx2_z00_z99', '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', 'label'],
      dtype='object')
```

Figure 18 Informasi mengenai 53 fitur pada dataset

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 untuk memeriksa apakah terdapat nilai yang hilang (*missing*)

```
[ ]  #checking null value
     C = (df.dtypes == 'object')
     CategoricalVariables = list(C[C].index)

     Integer = (df.dtypes == 'int64')
     Float = (df.dtypes == 'float64')
     NumericVariables = list(Integer[Integer].index) + list(Float[Float].index)

     Missing_Percentage = (df.isnull().sum()).sum()/np.product(df.shape)*100
     print("The number of missing entries before cleaning: " + str(round(Missing_Percentage,5)) + " %")

     The number of missing entries before cleaning: 0.0 %
```

Figure 19 Potongan kode untuk melihat *missing value*

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.

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:

```
[ ]  C = (df.dtypes == 'object')
     C2 = (df.dtypes == 'category')
     CategoricalVariables = list(C[C].index) + list(C2[C2].index)

     Integer = (df.dtypes == 'int64')
     Float = (df.dtypes == 'float64')
     NumericVariables = list(Integer[Integer].index) + list(Float[Float].index)

     df_kategori = pd.get_dummies(df[CategoricalVariables], columns=CategoricalVariables)
     df_numeric = df[NumericVariables]

     df_dummy = pd.get_dummies(df[CategoricalVariables], columns=CategoricalVariables)
     df_numeric = df[NumericVariables]

     df_numeric["id"] = df_numeric.index + 1
     df_dummy["id"] = df_dummy.index + 1
     bpjs_data_final = pd.merge(df_dummy, df_numeric ,on="id")
     bpjs_data_final.drop(['id'], axis=1, inplace=True)
     print("Dummy transformation was successful")
```

Figure 20 Transformasi atribut kategorik menjadi numerik

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
```

Figure 21 Pengecekan atribut fitur setelah transforamasi

3.4 Binning

Tahapan ini merupakan proses transformasi data dengan menggunakan metode *binning.* Metode ini akan digunakan untuk mengelompokkan data numerik menjadi beberapa bin dengan tujuan memudahkan pemahaman pada persebaran data yang digunakan. Berdasarkan analisis yang didapatkan, diketahui bahwa fitur **umur** dan **LoS** merupakan nilai bertipe numerik dan memiliki persebaran data yang tidak merata. Oleh karena itu akan dilakukan proses *binning* pada kedua fitur tersebut.

Untuk fitur umur akan dibagi menjadi 5 kategori dengan bin yang ditentukan adalah sesuai dengan kategori usia berdasarkan WHO yaitu sebagai berikut.

Bin 1: umur <=1,

Bin 2: 2<= umur >=10,

Bin 3: 11<= umur >=19,

Bin 4: 20<= umur >=60,

Bin 5: umur > 60

```
[ ]
    # binning dataset
    import numpy as np
    import math
    from sklearn.datasets import load_iris
    from sklearn import datasets, linear_model, metrics

    batas_bin = [ -1, 2, 11, 20, 61, 120]
    kategori = ['satu', 'dua', 'tiga', 'empat', 'lima']
    df['umur'] = pd.cut(df['umur'], bins=batas_bin, labels=kategori)
```

Figure 22 Binning untuk fitur Umur

Untuk fitur los yang memiliki hubungan terhadap jnpsplesep yang terkait pada tipe rawat inap atau rawat jalan selanjutnya akan dikelompokkan menjadi 4 kategori yaitu ‘rawat jalan’, ‘short stay’, ‘medium stay’, ‘long stay’. Penentuan bin adalah sebagai berikut.

los = 0

0 : rawat jalan,

1-5 : short stay,

6- 10 : medium stay,

> 10 : long stay

```
[ ]  # binning dataset
     import numpy as np
     import math
     from sklearn.datasets import load_iris
     from sklearn import datasets, linear_model, metrics

     batas_bin = [-1, 1, 6, 11,800 ]
     kategori = ['rawat jalan', 'short stay', 'medium stay', 'long stay']
     df['los'] = pd.cut(df['los'], bins=batas_bin, labels=kategori)
```

Figure 23 Binning untuk fitur los

3.5 Standardization

Untuk tahap ini akan dilakukan standarisasi pada data yang telah diolah sebelumnya untuk mendapatkan hasil yang lebih baik. Sebelum standarisasi dijalankan, terlebih dahulu data dibagi dan disimpan dalam variabel X dan y seperti dibawah ini.

```
X = data.iloc[:, 0:-1]
y = data.iloc[:, -1]
```

Figure 24 Pembagian dan penyimpanand data dalam variabel X dan y

Penerapan standarisasi berfokus pada mengubah data mentah menjadi informasi yang dapat digunakan sebelum dianalisis. Merupakan teknik yang menskalakan data sehingga memiliki mean = 0 dan standar deviasi =1.

```
[ ]  # standardization
     from numpy import asarray
     from sklearn.preprocessing import StandardScaler

     # define standard scaler
     scaler = StandardScaler()
     # transform data
     X = scaler.fit_transform(X)
     print(X)
```

Figure 25 Standarisasi fitur

Dan output yang diperoleh adalah sebagai berikut:

```
[[-0.4873673  -0.51753857 -0.17502359 ... -0.65076355 -0.09649292
   0.        ]
 [-0.4873673   1.93222313 -0.17502359 ...  2.42227887 -0.09649292
   0.        ]
 [ 2.05184056 -0.51753857 -0.17502359 ... -0.65076355 -0.09649292
   0.        ]
 ...
 [-0.4873673  -0.51753857 -0.17502359 ... -0.65076355 -0.09649292
   0.        ]
 [ 2.05184056 -0.51753857 -0.17502359 ...  0.11749706 -0.09649292
   0.        ]
 [-0.4873673  -0.51753857 -0.17502359 ... -0.65076355 -0.09649292
   0.        ]]
```

BAB 4
MODELLING

Tahap keempat pada metodologi CRISP-DM untuk melakukan binary classification dalam mendeteksi fraud adalah modeling. Pada bab ini akan dijelaskan mengenai pemilihan teknik modelling, dan menghasilkan *test scenario* serta teknik membangun model yang akan dibangun.

4.1 Build Test Scenario

Pada proses melakukan *data mining*, pemilihan model akan dipengaruhi oleh tujuan dari pelaksanaannya. Sebelum melakukan pembangunan model, perlu dilakukan perancangan bagaimana model yang akan dibangun. Analisis melalui pengujian model yang akan dipilih yaitu sebagai berikut.

1. Model menggunakan seluruh features

   Pada model ini, akan dibangun menggunakan seluruh features pada dataset. Sebelumya diketahui terdapat 53 features sebelum dilakukan *data preprocessing*. Pada model ini akan dilakukan prediksi menggunakan RandomForestClassification. Prediksi yang dilakukan menghasilkan akurasi untuk data train dan data test masing-masing sebesar 0.93 dan 0.67

2. Model menggunakan best features

   Pada teknik pemodelan berikutnya, dilakukan pemilihan *best features* dengan memanfaatkan fungsi SelectKBest dengan total K sebesar 70. Pada model ini akan dilakukan prediksi menggunakan RandomForestClassification. Prediksi yang dilakukan menghasilkan akurasi untuk data train dan data test masing-masing sebesar 0.85 dan 0.67

3. Model menggunakan fitur ['kdkc', 'typeppk', 'jkpst', 'umur', 'jnspelsep', 'los', 'cmg', 'severitylevel', 'diagprimer', 'label']

Pada pemodelan ini akan dilakukan dengan memilih hanya feature tertentu untuk digunakan sebagai data test maupun data train. Untuk itu, features yang tidak digunakan akan di drop sesuai dengan kebutuhan. Adapun features yang akan digunakan adalah 'kdkc', 'typeppk', 'jkpst', 'umur', 'jnspelsep', 'los', 'cmg', 'severitylevel', 'diagprimer', 'label'. Pada model ini akan dilakukan prediksi menggunakan RandomForestClassification. Prediksi yang dilakukan menghasilkan akurasi untuk data train dan data test masing-masing sebesar 0.89 dan 0.67

4. Model dengan menggunakan tuning hyperparameter pada fitur yang dipilih

   Pada pemodelan ini akan dilakukan teknik tuning hyper parameter yaitu kita dapat melakukan pengaturan pada algoritma dengan mengubah parameter untuk menemukan kinerja yang optimal. Jumlah *estimator* akan ditunjukkan dengan nilai *start* dan *stop* yang telah ditentukan, kemudian hyperparameter akan ditemukan dengan menggunakan fungsi random_grid. Hasil hyperparameter tersebut kemudiana akan digunakan sebagai parameter prediksi dengan fungsi RandomizedSearchCV. Prediksi yang dilakukan menghasilkan akurasi untuk data train dan data test masing-masing sebesar 0.81 dan 0.68

4.2 Model Building

Berdasarkan pengujian untuk model yang telah ditemukan sebelumnya, maka pada proyek ini akan menggunakan model dengan tuning hyperparameter untuk *features* yang dipilih. Hal ini berdasarkan hasil yang diperoleh dari akurasi untuk *data train* dan *data test* yang menunjukkan *overfitting* yang lebih kecil dibandingkan dengan model lainnya. Dalam pembangunan model klasifikasi terdapat 3 informasi yang perlu didefinisikan untuk kemudian digunakan dalam pengambilan keputusan dalam *data mining*, antara lain:

a. Parameter *settings,* digunakan untuk penentuan parameter yang akan digunakan pada model

b. Membuat model menggunakan algoritma yang sudah ditentukan

c. Menampilkan hasil penilaian akurasi terhadap data latih dan data uji yang dimiliki

Binary Classification dengan algoritma RFC dibangun pada bahasa pemrograman *python* dengan memanfaatkan *library python* yaitu *scikit-learn*. *Scikit-learn* merupakan salah satu *library* yang disediakan *python* untuk membangun model machine learning seperti regresi, *clustering* dan *classification*. Pada tahap pemodelan ini, dataset yang digunakan merupakan dataset yang telah diproses sebelumnya seperti yang sudah dijelaskan pada bab 2 dan 3. Untuk pengimplementasian model RFC, tahap pertama yang dilakukan adalah membagi 2, yaitu: data latih dan data uji dengan persentase 70% untuk data latih dan 30% untuk data uji. Data latih akan digunakan untuk membangun model dan data uji akan digunakan untuk menguji model yang telah dibangun.

```
# implementing train-test-split

from sklearn.model_selection import KFold, cross_val_score, train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state = 60)
print(X_train.shape)
```

```
(140151, 119)
```

Figure 26 Implementasi untuk membagi data menjadi data latih dan data uji

Kemudian selanjutnya dilakukan pendefinisian 3 informasi yang dibutuhkan dalam pembuatan keputusan *data mining*, yaitu:

a. Parameter *settings,* digunakan untuk penentuan parameter yang akan digunakan pada model

   Berdasarkan pengujian parameter yang telah dilakukan dengan parameter tuning, diperoleh kesimpulan bahwa parameter `random_state=5, n_estimators=20` menghasilkan pemodelan dengan akurasi terbaik. Maka pada pemodelan RFC parameter random_state = 5 dan n_estimators = 20 akan digunakan.

```
# random forest model creation
rfc =  RandomForestClassifier(random_state=5, n_estimators=50, )
```

Figure 27 Parameter setting

b. Membuat model menggunakan algoritma yang sudah ditentukan

   Selanjutnya adalah pembangunan model berdasarkan algoritma yang dipilih yaitu RFC. Untuk pembangunan model sendiri menggunakan potongan kode berikut ini:

```
from sklearn import model_selection
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc_auc_score
# random forest model creation
rfc =  RandomForestClassifier(random_state=5, n_estimators=50, )
rfc.fit(X_train,y_train)
# predictions
rfc_predict = rfc.predict(X_test)
```

Figure 28 Pemodelan dengan RFC

c. Menampilkan hasil penilaian akurasi terhadap data latih dan data uji yang dimiliki

```
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report, confusion_matrix
train_accuracy= rfc.score(X_train, y_train)
test_accuracy= rfc.score(X_test, y_test)
print(train_accuracy)
print(test_accuracy)
```

```
0.9392726416507909
0.6764059534512037
```

Figure 29 Hasil akurasi data latih dan data uji

BAB 5
MODEL EVALUATION

Pada bab ini akan dijelaskan mengenai evaluasi terhadap model pendeteksi potensi kecurangan pada layanan BPJS yang dihasilkan menggunakan algoritma *Random Forest Classification.* Evaluasi adalah fase interpretasi terhadap hasil *data mining*. Evaluasi dilakukan secara mendalam dengan tujuan agar hasil pada tahap *modelling* sesuai dengan sasaran yang ingin dicapai.

5.1 Evaluation of Modeling Result

Tahap ini dilakukan untuk mengetahui performa *binary classification* untuk mendeteksi *fraud* menggunakan *confusion matrix* dan *classification report* berdasarkan dataset yang digunakan yaitu data BPJS Kesehatan yang berasal dari dataset yang digunakan dalam kompetisi Hackathon. Sebelum pengerjaan proyek telah ditetapkan serangkaian ketentuan/ standar akurasi *precision, recall* dan *accuracy* pembangunan model. Dimana score *precision* > 0.54, *recall* > 0.65 dan *accuracy* > 0.56. Pada tahap pembangunan model, telah dilakukan penilaian akurasi terhadap data latih dan data uji. Dan pada tahap ini dilakukan evaluasi pemodelan dengan melihat *precision, recall* dan *accuracy* yang dilakukan adalah sebagai berikut:

```
=== Confusion Matrix ===
[[20107  9814]
 [ 8940 21205]]


=== Classification Report ===
              precision    recall  f1-score   support

           0       0.69      0.67      0.68     29921
           1       0.68      0.70      0.69     30145

    accuracy                           0.69     60066
   macro avg       0.69      0.69      0.69     60066
weighted avg       0.69      0.69      0.69     60066
```

Berdasarkan hasil yang diperoleh dari pembangunan model dengan menggunakan algoritma RFC telah menghasilkan model dengan akurasi cukup baik dengan score > 0.5 dan yang

memenuhi standar dan ketentuan pembangunan proyek. Model yang dibangun telah cukup baik dalam menerapkan algoritma RFC untuk mendeteksi kecurangan pada layanan BPJS. Selanjutnya evaluasi dilanjutkan dengan melakukan pemetaan kesesuaian output dari model menggunakan visualisasi heatmap, dan diperoleh hasil sebagai berikut:

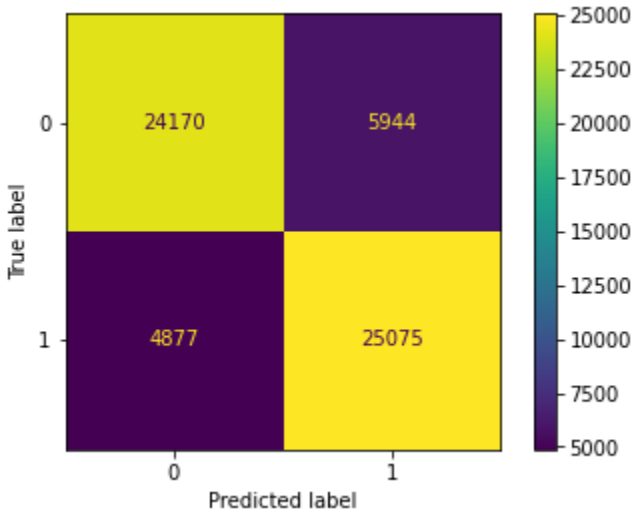


Figure 30 Visualisasi hasil evaluasi dengan heatmap

Karena penelitian ini merupakan *binary classification*, maka output akhir dari pemodelan ini adalah binary [0, 1], dimana 0 berarti terdapat tidak *fraud* dan 1 artinya terdapat *fraud*. Berdasarkan heatmap yang diperoleh dapat dilihat hubungan *predicted lable* dengan *true lable* dalam menghasilkan data *valid* dan tidak *valid*. Data valid yang diperoleh berupa: data yang diprediksi tidak *fraud* dan benar tidak *fraud* berjumlah 24170 dan data yang diprediksi *fraud* dan benar *fraud* berjumlah 25075. Sementara untuk data tidak *valid* yang diperoleh berupa: data yang diprediksi tidak *fraud* tetapi kebenarannya adalah *fraud* berjumlah 5944 dan data yang diprediksi *fraud* tetapi kebenarannya adalah tidak *fraud* berjumlah 4877.

5.2 Modeling Process Review

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

- Proses eksplorasi data akan membantu dalam memilih atribut yang berkaitan dengan mendeteksi terjadinya *fraud* pada layanan BPJS.

- *Data Preparation*, khususnya pada proses data *cleaning* dan *transform*, sehingga data yang diperoleh dapat menghasilkan model yang baik.
- Sangat penting untuk tetap fokus pada masalah bisnis yang dihadapi, karena setelah data siap dianalisis, maka akan dilakukan tahap pemodelan. *Business understanding* sangat penting dalam memutuskan bagaimana menerapkan hasil yang diperlukan dalam mendeteksi terjadinya *fraud* pada layanan BPJS.

5.3 Determine Next Step

Tahapan ini menentukan langkah apa yang akan diambil selanjutnya. Berdasarkan hasil evaluasi terhadap model yang digunakan dengan algoritma RFC, dengan hasil akurasi pemodelan yang diperoleh dalam mendeteksi terjadinya *fraud* pada layanan BPJS, maka diputuskan pengerjaan proyek akan dilanjutkan ke tahap akhir yakni deployment.

BAB 6
DEPLOYMENT

Tahap keenam pada metodologi CRISP-DM untuk melakukan prediksi kinerja karyawan adalah deployment. Pada bab ini akan dijelaskan mengenai perencanaan dan *deployment* model yang sudah dihasilkan, serta laporan akhir untuk proses *data mining* yang sudah dilakukan.

6.1 Model Deployment

Model yang sudah selesai dibangun selanjutnya dilanjutkan pada tahap *deployment.* Model *deployment* merupakan proses dimana model yang telah dibangun akan tersedia pada lingkungan produksi dimana model tersebut dapat melakukan prediksi pada sistem lain. Model *deployment* yang dilakukan pada proyek ini adalah berdasarkan pola secara dinamis yang akan di-*deploy* pada web browser, sehingga akan ditampilkan dalam bentuk website. *Deployment* model yang telah dibangun akan dilakukan pada aplikasi Heroku yaitu salah satu tools yang termasuk pada *Platform As A Service (PaaS)* untuk mengelola dan menjalankan aplikasi dari model yang dikembangkan. Aplikasi tersebut akan diterapkan pada Heroku dengan menggunakan Flask Python

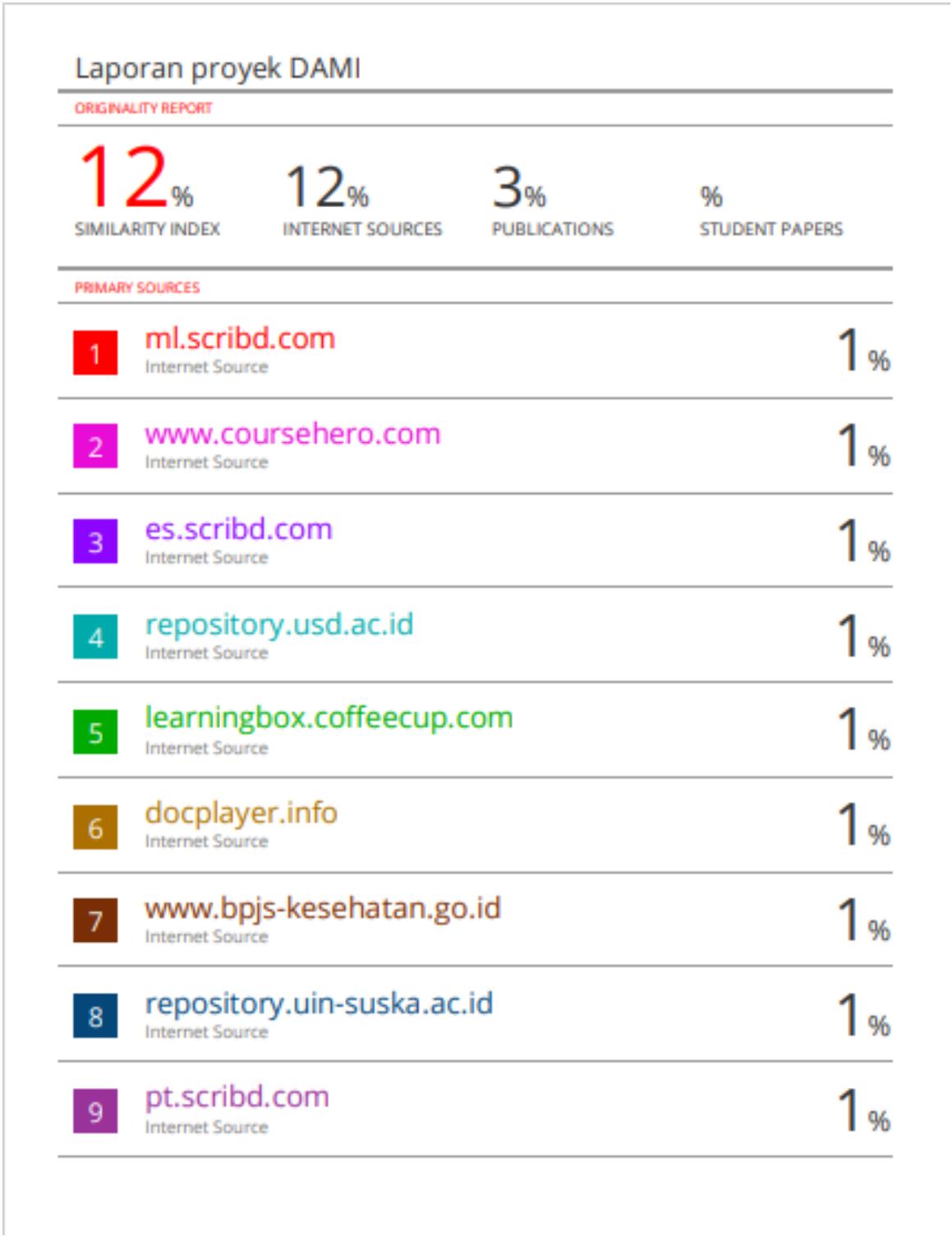
6.2 Final Report

Selama pengerjaan proyek ini, anggota tim terlibat dan berkontribusi dalam pengerjaan proyek dari awal hingga tahapan selesai dilakukan. Hal ini menjadi sarana pembelajaran bagi anggota tim menerapkan dalam dunia nyata pelaksanaan *data mining* sesuai dengan tahapan CRISP-DM. Tim juga dapat memahami dan mampu bereksplorasi pada data, tahap pemrosesan data, penerapan algoritma dalam membangun model, melakukan evaluasi untuk menilai performa model, hingga melakukan *deployment* untuk model yang telah dibangun. Tahapan yang dilakukan tim proyek setelah melakukan *deployment* adalah membuat dokumentasi yang dituangkan dalam laporan akhir. Laporan akhir mencakup penjelasan terkait dengan rangkaian proses *data mining* yang dilakukan sesuai dengan metodologi CRISP-DM yaitu dimulai dari *business understanding, data understanding, data preparation, modelling, evaluation* hingga *deployment.* Terdapat *deliverables* lain yang akan dihasilkan dari pelaksanaan proyek ini yaitu video presentasi, poster, dan *model deployment*

yang disajikan melalui aplikasi heroku. *Deliverables* yang dihasilkan akan menyampaikan semua tahapan hingga hasil dari pengerjaan proyek ini.

LAMPIRAN

Berikut merupakan tangkapan layar untuk hasil cek turnitin dari dokumen Laporan Akhir_12S18018_12S18019_12S18064:



Laporan proyek DAMI

ORIGINALITY REPORT

12%	12%	3%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ml.scribd.com Internet Source	1%
2	www.coursehero.com Internet Source	1%
3	es.scribd.com Internet Source	1%
4	repository.usd.ac.id Internet Source	1%
5	learningbox.coffeecup.com Internet Source	1%
6	docplayer.info Internet Source	1%
7	www.bpjs-kesehatan.go.id Internet Source	1%
8	repository.uin-suska.ac.id Internet Source	1%
9	pt.scribd.com Internet Source	1%