# LAPORAN PROYEK DATA MINING

## *Fraud Detection with Binary Classification Using ANN Algorithm*


**Disusun oleh:**

12S19001 – Raynaldo Silalahi

12S19009 – Manuel Sigalingging

12S19040 – Abel M. Y. Tampubolon

**PROGRAM STUDI SARJANA SISTEM INFORMASI**

**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO INSTITUT TEKNOLOGI DEL**

**NOVEMBER 2022**

# DAFTAR ISI

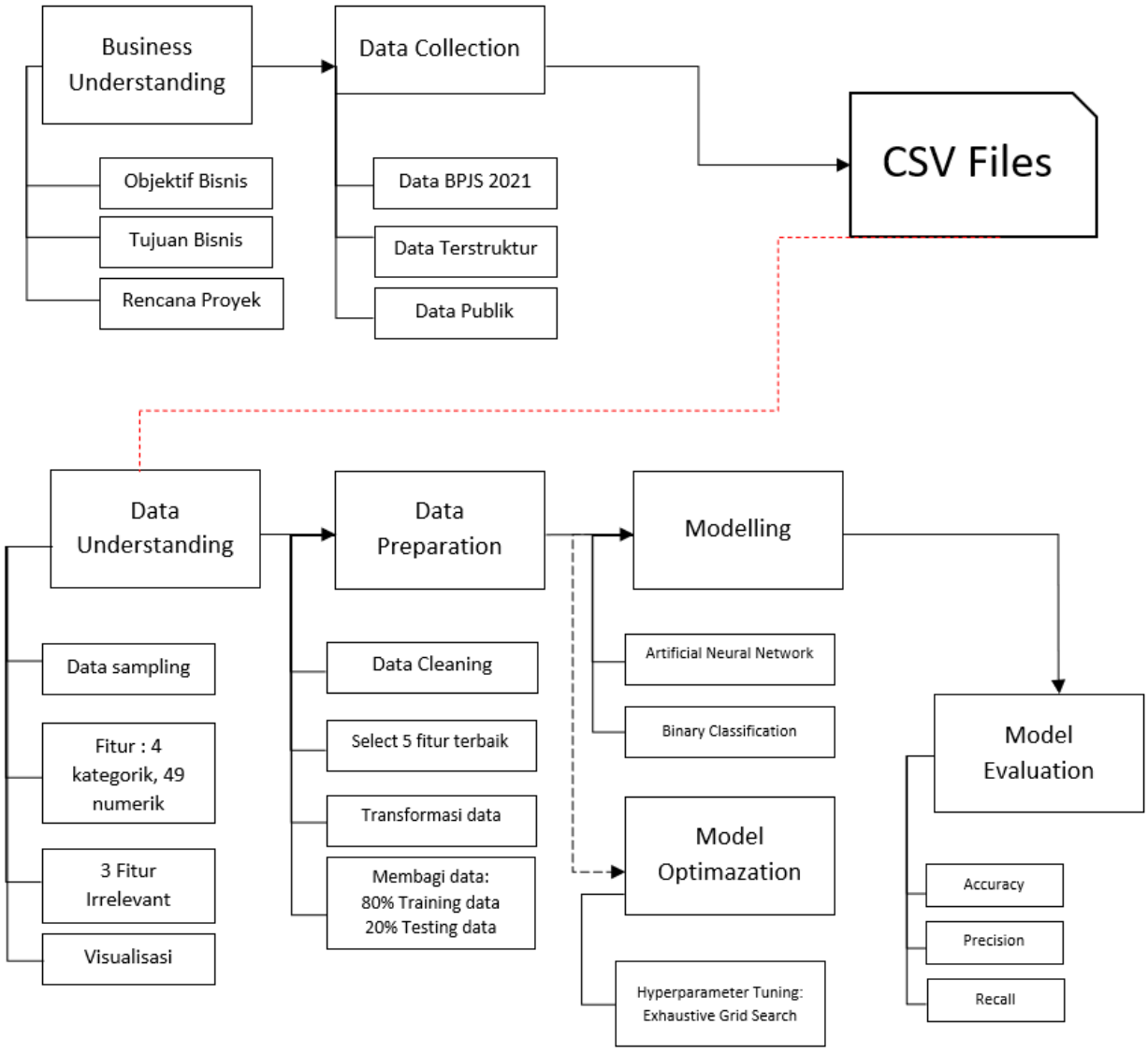
DAFTAR ISI....2
DAFTAR GAMBAR....3
PENDAHULUAN....4
BUSINESS UNDERSTANDING....6
1.1 Menentukan Objektif Bisnis....6
1.2 Menentukan Tujuan Bisnis....7
1.3 Rencana Pengerjaan Proyek....7
DATA UNDERSTANDING....9
2.1 Mengumpulkan dan Menelaah Data....9
2.2 Deskripsi Data....9
2.3 Telaah Data....11
2.4 Memvalidasi Data....17
DATA PREPARATION....18
3.1 Selecting Data....18
3.2 Data Cleaning....19
3.3 Construct Data....19
3.4 Binning....20
3.5 Standardization....21
MODELLING....24
4.1 Build Test Scenario....24
4.2 Build Modeling....24
MODEL EVALUATION....27
5.1 Evaluation of Modelling Result....27
5.2 Modelling Process Review....28
DEPLOYMENT....29
LAMPIRAN....30

# DAFTAR GAMBAR

Gambar 1.....4
Gambar 2.....12
Gambar 3.....12
Gambar 4.....12
Gambar 5.....13
Gambar 6.....13
Gambar 7.....13
Gambar 8.....14
Gambar 9.....14
Gambar 10.....15
Gambar 11.....16
Gambar 12.....16
Gambar 13.....17
Gambar 14.....18
Gambar 15.....18
Gambar 16.....19
Gambar 17.....19
Gambar 18.....20
Gambar 19.....21
Gambar 20.....21
Gambar 21.....22
Gambar 22.....22
Gambar 23.....25
Gambar 24.....25
Gambar 25.....25
Gambar 26.....26
Gambar 27.....27
Gambar 28.....28

# PENDAHULUAN

Salah satu jenis cara dalam melakukan klasifikasi ialah Binary Classification. Binary Classification merupakan proses mengklasifikasikan dalam elemen yang membagi menjadi dua kelas antara normal dan abnormal (0 dan 1). Perencanaan proyek yang akan dibahas mengenai fraud detection terhadap data BPJS digunakan algoritma *Artificial Neural Network* atau ANN dalam melakukan klasifikasi. Metodologi pengerjaan proyek yang digunakan yakni metodologi CRISP-DM. Untuk tahapan terhadap pengerjaan proyek dengan metode CRISP-DM dapat dilihat pada gambar berikut:




*Gambar 1*

Tahapan dalam pengerjaan proyek terdiri atas *business understanding* dimana menentukan objektif bisnis, tujuan bisnis, dan rencana pengerjaan proyek. Lalu *data collection*, pengumpulan dataset yang akan digunakan dalam membangun model. Selanjutnya *Data*

*preparation*, untuk mempersiapkan data yang akan digunakan berupa membersihkan dan mentransformasi data. *Modelling* untuk membangun model klasifikasi dan *Model Evaluation* untuk mengevaluasi hasil akurasi dari model yang dibangun. Apabila model tercapai dibangun dan memenuhi target akurasi yang diinginkan, maka model akan dibangun pada tahap *Deployment Mode.*

# BUSINESS UNDERSTANDING

Proses awal dalam metodologi yang digunakan pada CRISP-DM diawali dengan business understanding, yang dimana pada proses tahapan ini memberikan penjelasan tentang objektif bisnis, tujuan bisnis, dan rencana pengerjaan proyek.

## 1.1 Menentukan Objektif Bisnis

BPJS singkatan dari Badan Penyelenggara Jaminan Sosial merupakan sebuah badan hukum yang memiliki tujuan dalam memberikan jaminan mengenai perlindungan dan pelayanan kesehatan terhadap penggunanya. Untuk setiap jaminan kesehatan seseorang menjadi jaminan perlindungan bagi mereka yang telah membayarkan iuran wajib atau subsidi dari pemerintah kepada pihak BPJS. Iuran atau pembayaran yang dilakukan berdasarkan kelompok atau golongan kepesertaannya dalam BPJS.

Tetapi, munculnya masalah dalam pengelolaan dana pada BPJS yang kurang efektif mengenai jaminan sosial BPJS Kesehatan Indonesia yang diberikan. Pembayaran yang telah dilakukan tidak sesuai dengan daftar ketentuan yang telah ditetapkan oleh BPJS. Penyebab terjadinya disebabkan oleh oknum tertentu dengan melakukan manipulasi pengolahan data demi kepentingan sendiri dan peningkatan kerugian BPJS secara cepat.

Fraud merupakan suatu kegiatan yang sering terjadi di mana saja. Seluruh tindak *fraud/*penipuan tentu merugikan pihak yang menjadi korban. Pencegahan dalam *fraud/*penipuan harus diidentifikasi dengan serius agar mampu mengetahui pihak-pihak yang terjaring dalam penipuan. Salah satu tindak kejahatan *fraud/*penipuan yang terjadi pada BPJS menjadi adanya potensi yang menimbulkan kerugian kepada BPJS. Dalam menghindari timbulnya kerugian yang sangat besar tersebut, tentu dilakukannya upaya pencegahan. Salah satu upaya pencegahan yang mampu dilakukan yakni dengan menganalisis apa yang menjadi faktor yang mempengaruhi terjadinya *fraud* dan dilakukannya prediksi terhadap timbulnya terjadi penipuan dengan menggunakan *data mining*.

Dalam proyek ini mengubah klaim asuransi kesehatan pada pelayanan kesehatan di rumah sakit sebagai titik fokus untuk memprediksi *fraud,* sehingga objektif pada proyek ini yaitu:

- Melakukan prediksi potensi terjadinya *fraud* pada klaim pelayanan rumah sakit
- Mencari faktor-faktor yang mempengaruhi terjadinya penipuan

Proyek ini dapat dikatakan berhasil jika:

- Ditemukan prediksi terhadap potensi terjadinya *fraud* pada klaim pelayanan rumah sakit
- Ditemukan faktor-faktor yang mempengaruhi terjadinya *fraud*

### 1.2 Menentukan Tujuan Bisnis

Penipuan yang terjadi menyebabkan kerugian yang dapat mengancam keberadaan bisnis menjadi berubah. Maka, tujuan bisnis dari pengerjaan proyek ini ialah melakukan prediksi terhadap munculnya atau terjadinya penipuan/*fraud* terhadap klaim dalam pelayanan BPJS Kesehatan dan melakukan evaluasi terhadap faktor yang mempengaruhi terhadap penipuan atau *fraud* dengan mengidentifikasi gambaran relasi antar data. Dengan mengetahui gambaran yang diperoleh, manfaat yang dapat diberikan dari penelitian ini dapat berguna untuk melakukan evaluasi proses terhadap klaim pelayanan BPJS rumah sakit ke depannya agar tujuan bisnisnya sesuai dengan yang diharapkan.

### 1.3 Rencana Pengerjaan Proyek

Pada tahapan perencanaan yang dilakukan dalam proyek ini mengacu pada Standar Kompetensi Kerja Nasional: KepMen Ketenagakerjaan No 299 tahun 2020 dengan metodologi CRISP-DM. Berikut tahapan perencanaan dalam pengerjaan proyek.

| **Aktivitas** | **Detail** | **Waktu (Minggu Ke-)** | | | | |
|---|---|---|---|---|---|---|
| | | **12** | **13** | **14** | **15** | **16** |
| Pemilihan Kasus dan Algoritma | Pemilihan Kasus | | | | | |
| | Penentuan Algoritma | | | | | |
| Business Understanding | Menentukan Objektif Bisnis | | | | | |
| | Menentukan Tujuan Bisnis | | | | | |
| | Membuat Rencana Proyek | | | | | |
| Data Understanding | Mengumpulkan Data | | | | | |
| | Mendeskripsikan dan Menelaah Data | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Memvalidasi Data | | | | | |
| Data Preparation | Memilah Data | | | | | |
| | Membersihkan Data | | | | | |
| | mengkonstruksi Data | | | | | |
| | Menentukan Label Data | | | | | |
| | Mengintegrasikan Data | | | | | |
| Modelling | Membangun Skenario Pengujian | | | | | |
| | Membangun Model | | | | | |
| Model Evaluation | Mengevaluasi Hasil Pemodelan | | | | | |
| | Melakukan Review Proses Pemodelan | | | | | |

Dalam mendukung pada pengerjaan proyek untuk setiap tahapan, berikut spesifikasi dalam development environment yang digunakan untuk proyek ini.

- Tools : 1. Jupyter Notebook
  2. Google Collab
- Bahasa Pemrograman : Python
- Algoritma : Artificial Neural Network

# DATA UNDERSTANDING

## 2.1 Mengumpulkan dan Menelaah Data

Tahap kedua pada metodologi CRISP-DM adalah *data understanding*, dimana pada tahap ini akan mendapatkan informasi mengenai kekurangan dan kelebihan data, tingkat kesesuaian data dengan bisnis yang akan dipecahkan, hingga ketersediaan data. Tahapan ini diperlukan karena dataset belum tentu bisa langsung digunakan.

## 2.2 Deskripsi Data

*Dataset* yang digunakan untuk *fraud detection* BPJS Kesehatan 2021 adalah fraud_detection_train *dataset*. Dataset ini berisi 53 atribut dan 200217 *rows*. Berikut tabel deskripsi setiap atributnya:

| No. | Atribut | Keterangan | Tipe | Nilai |
|---|---|---|---|---|
| 1 | visit_id | Id kunjungan | int64 | ID numerik |
| 2 | kdkc | Kode wilayah kantor cabang BPJS Kesehatan | int64 | Kode numerik |
| 3 | dati2 | Kode kabupaten/kota | int64 | Kode numerik |
| 4 | typeppk | Kode tipe rumah sakit | object | SC, C, B, SD, SB, A, ,D, I3, KM, KI, I2, I4, KJ, KL, I1, KB, KC, GD, SA, KP, KO, KG, HD, KT, KU |
| 5 | jkpst | Jenis kelamin peserta JKN-KIS | object | P (Perempuan), L (Laki- |

| | | | | |
|---|---|---|---|---|
| | | | | Laki) |
| 6 | umur | Umur peserta saat mendapatkan pelayanan di rumah sakit | int64 | 0-109 |
| 7 | jnspelsep | Tingkat layanan: 1 (rawat inap), 2 (rawat jalan) | int64 | 1: Rawat Inap<br>2: Rawat Jalan |
| 8 | los | Lama peserta dirawat di rumah sakit | int64 | 0-255 |
| 9 | cmg | Klasifikasi CMG (Case Mix Group) | object | 'F','E','Q','L','H','W','P','U','K','G','M','N', 'A','C', 'D', 'Z', 'J', 'O', 'S', 'I', 'V', 'T', 'B' |
| 10 | severitylevel | Tingkat urgensi | int64 | 0, 1, 2, 3 |
| 11 | diagprimer | Diagnosa primer | object | 'f00_f99', 'e00_e90', 'r00_r99', 'j00_j99', 's00_t98', 'h00_h59', 'm00_m99', 'c00_d48', 'z00_z99', 'p00_p96', 'h60_h95', 'k00_k93', 'g00_g99', 'i00_i99', |

| | | | | |
|---|---|---|---|---|
| | | | | 'l00_l99', 'a00_b99', 'n00_n99', 'o00_o99','d50_d89', 'q00_q99', 'u00_u85' |
| 12 | dx2_…_… | Diagnosa Sekunder | int64 | terdapat 22 atribut dimana nilainya adalah 0 - 13 |
| 13 | proc…_ | Kode Procedure | int64 | terdapat 19 atribut dimana nilainya adalah 0 - 23 |
| 14 | label | Flag fraud: 1 (Fraud), 0 (Tidak Fraud) | int64 | 1:fraud; 0:tidak fraud |

### 2.3 Telaah Data

Exploratory Data Analysis merupakan salah satu pendekatan untuk memahami dan menganalisis data. Dengan melakukan EDA kita dapat mendapatkan informasi berupa konteks data, statistik dataset, keseimbangan dataset, tipe data pada tiap atribut, hingga kualitas data.

Hasil analisis EDA sebagai berikut:

- Melihat ukuran data dari dataset faud_detection_train.csv

```
In [3]: # View the data size
        data.shape

Out[3]: (200217, 53)
```

*Gambar 2*

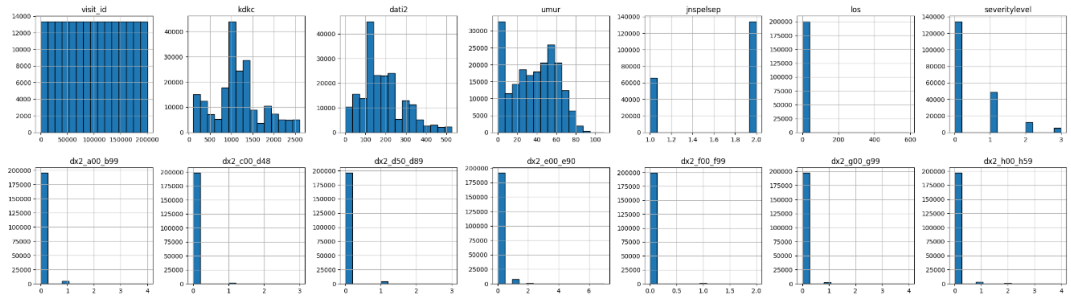
- Melihat daftar fitur atau variabel yang ada pada dataset. pada dataset ini ditemukan 53 fitur.

```
In [7]: data.columns

Out[7]: 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')
```

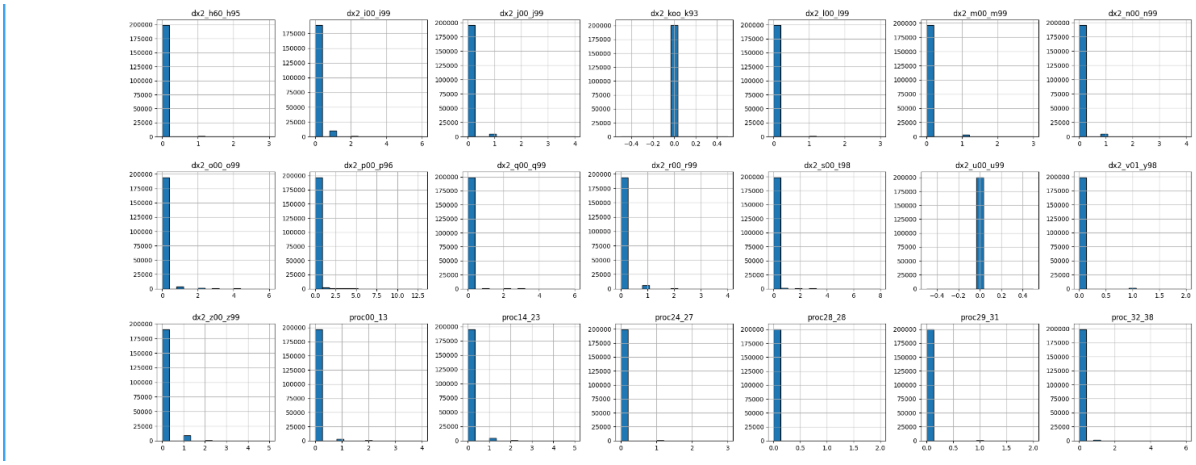
*Gambar 3*

- Menampilkan semua variabel pada dataset dengan histogram

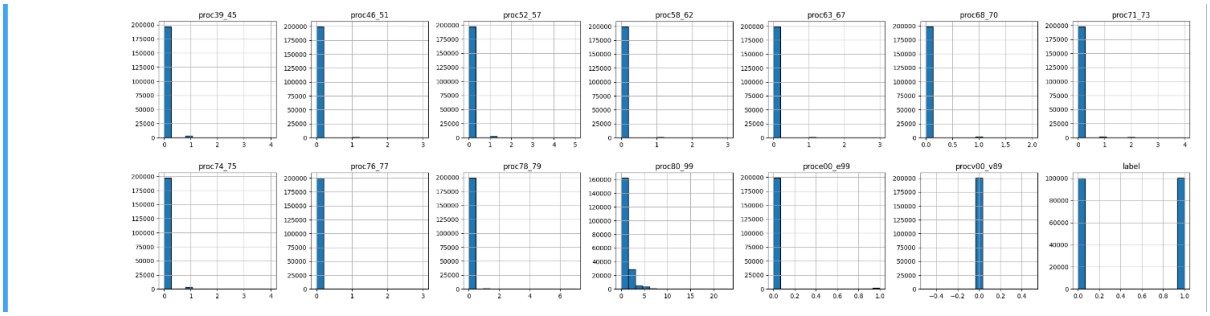
```
In [14]: ## Showing all variables in histogram
         data.hist(edgecolor = 'black', bins = 15, figsize = (30, 30));

Matplotlib is building the font cache; this may take a moment.
```




*Gambar 4*




*Gambar 5*




*Gambar 6*

- Memeriksa statistik dataset yang digunakan dengan menggunakan fungsi *data.describe()*. Adapun kode program beserta output sebagai berikut:

```
In [4]: data.describe()
```

Out[4]:

| | 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 | 200217.000000 |
| mean | 100109.000000 | 1147.367816 | 184.793309 | 36.850602 | 1.669778 | 1.303356 | 0.444003 | 0.024893 | 0.008341 | 0.02070 |
| std | 57797.813761 | 574.486224 | 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 7*

Output menampilkan rangkuman data set yang meliputi *count*, *mean*, *standard of deviation*, *minimum value*, *quantile* 1 (25%), *quantile* 2 (50%), *quantile* 3 (75%), dan *maximum value*.

- Memeriksa keseimbangan dataset yang digunakan. Adapun kode program yang digunakan sebagai berikut:

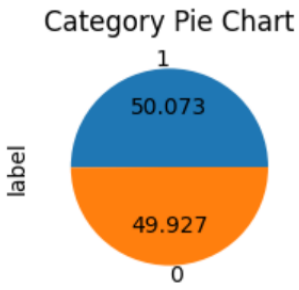
```
In [8]: All = data.shape[0]
        fraud = data[data['label'] == 1]
        nonFraud = data[data['label'] == 0]

        totalFraud = len(fraud)/All
        totalNonFraud = len(nonFraud)/All

        print('frauds :', totalFraud * 100, '%')
        print('non Frauds : ', totalNonFraud * 100, '&')

        frauds : 50.07317060988827 %
        non Frauds :  49.92682939011173 &
```

*Gambar 8*

- Memvisualisasikan keseimbangan data

```
import matplotlib.pyplot as plt

plt.figure(figsize=(2,2))
data['label'].value_counts().plot(kind='pie',autopct='%.3f')
plt.title('Category Pie Chart')
plt.show()

data['label'].value_counts()
```

✓ 1.8s




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

*Gambar 9*

- Memeriksa data tipe dari atribut pada dataset yang digunakan.

```
In [4]: data.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 10*

Fitur pada dataset yang digunakan terbagi menjadi 2 kategori yaitu 4 fitur non-numerik dan 49 fitur numerik.

- Memeriksa nilai null pada dataset dengan menggunakan fungsi notnull().Jika tidak ada data yang hilang, maka pada tabel akan menghasilkan nilai true. Kode program dan output yang digunakan adalah sebagai berikut:

```
In [12]: data.notnull()
```

Out[12]:

| | visit_id | kdkc | dati2 | typeppk | jkpst | umur | jnspelsep | los | cmg | severitylevel | ... | proc63_67 | proc68_70 | proc71_73 | proc74_75 | proc76_77 | proc78_79 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | True | True | True | True | True | True | True | True | True | True | ... | True | True | True | True | True | True |
| 1 | True | True | True | True | True | True | True | True | True | True | ... | True | True | True | True | True | True |
| 2 | True | True | True | True | True | True | True | True | True | True | ... | True | True | True | True | True | True |
| 3 | True | True | True | True | True | True | True | True | True | True | ... | True | True | True | True | True | True |
| 4 | True | True | True | True | True | True | True | True | True | True | ... | True | True | True | True | True | True |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 200212 | True | True | True | True | True | True | True | True | True | True | ... | True | True | True | True | True | True |
| 200213 | True | True | True | True | True | True | True | True | True | True | ... | True | True | True | True | True | True |
| 200214 | True | True | True | True | True | True | True | True | True | True | ... | True | True | True | True | True | True |
| 200215 | True | True | True | True | True | True | True | True | True | True | ... | True | True | True | True | True | True |
| 200216 | True | True | True | True | True | True | True | True | True | True | ... | True | True | True | True | True | True |

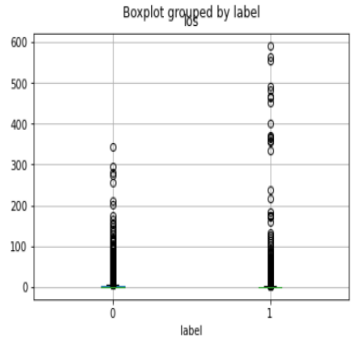
200217 rows × 53 columns

*Gambar 11*

- Memeriksa *outlier* pada dataset. *Outlier* yang ditemukan berdasarkan atribut los dan label dengan kode program dan output yang digunakan sebagai berikut:

```
In [11]: data.boxplot(column='los', by='label')
```

Out[11]: <AxesSubplot:title={'center':'los'}, xlabel='label'>




*Gambar 12*

- Memeriksa korelasi yang ada pada setiap atribut pada dataset dengan heatmap dan output yang dihasilkan adalah sebagai berikut:


*Gambar 13*

## 2.4 Memvalidasi Data

Pada tahapan validasi data dilakukan evaluasi untuk memastikan kualitas dan kelengkapan *dataset* yang digunakan pada proyek. Pada tahap ini dilakukan pengecekan kembali terhadap atribut yang tidak lengkap, pembersihan data (*data cleaning*) untuk memastikan data konsisten dan relevan, serta mengurangi jumlah dan kompleksitas data. Memperkirakan apakah semua value dan ejaan nilai-nilai rasional serta apakah fitur dengan value yang berbeda memiliki pengertian yang sama. Berikut adalah hasil yang ditemukan saat dilakukan penelusuran:

1. Dataset sudah seimbang dengan 100255 label *fraud* dan 99962 label tidak *fraud*;
2. Dataset terdiri atas 49 fitur numerik dan 4 fitur non-numerik;
3. Ditemukan 3 atribut yang hanya memiliki 1 nilai unik;
4. Tidak ditemukan data bernilai *null*; serta
5. Terdapat outlier pada atribut los.

# DATA PREPARATION

Data preparation merupakan tahapan ketiga pada metode ANN. Sebelum dilakukan pemodelan, data terlebih dahulu harus diperbaiki terlebih dahulu. Ada beberapa sub proses pada data preparation antara lain:

## 3.1 Selecting Data

Sebelum data digunakan dalam pemodelan, kita terlebih dahulu menyiapkan data dengan baik. Ada beberapa atribut yang tidak diperlukan dalam pemodelan nanti. Atribut tersebut akan di drop agar data menjadi lebih efisien.

```
# Selecting data
# Drop unused attribute

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

✓ 0.1s

*Gambar 14*

Berikut adalah potongan tampilan data setelah beberapa atribut didrop.

```
data.info()
```

✓ 0.1s

```
Output exceeds the size limit. Open the full output data in a text editor
<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
```

*Gambar 15*

```
data.columns
```
✓ 0.3s

```
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', 'proce00_e99',
       'label'],
      dtype='object')
```

*Gambar 16*

## 3.2 Data Cleaning

Data yang dipilih kemungkinan belum bersih sehingga dibutuhkan proses pembersihan data. Proses yang dilakukan pada data *cleaning* adalah menghapus objek data yang tidak memiliki atau mengandung nilai (*missing value)* dan menghapus atau mengeliminasi atribut yang tidak relevan.

```
# Checking missing values

A = (data.dtypes == 'object')
CategoricalVariables = list(A[A].index)

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

Missing_Percentage = (data.isnull().sum()).sum()/np.product(data.shape)*100
print("Total nilai yang missing sebelum dibersihkan: " + str(round(Missing_Percentage,5)) + "%" )
```
✓ 0.6s

```
Total nilai yang missing sebelum dibersihkan: 0.0%
```

*Gambar 17*

Berdasarkan output pengecekan *missing value*, dapat ditarik kesimpulan bahwa tidak ada atribut yang memiliki nilai *null*. Proses selanjutnya setelah ini adalah mengeliminasi objek data yang tidak relevan.

## 3.3 Construct Data

Pada tahap ini, dilakukan transformasi dengan tipe kategorik menjadi fitur numerik. Sehingga pada tahap data *construction* dilakukan agar data kemudian dapat di normalisasi. Tahap pertama yang dilakukan adalah pengecekan tipe data pada dataset dengan menggunakan fungsi *data.info()*.

```
In [21]: data.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
```

*Gambar 18*

Dikarenakan ada 4 atribut yang memiliki tipe data kategorik, perlu dilakukan transformasi data pada tipe data atribut dengan menjalankan potongan kode:

### 3.4 Binning

Pada tahap ini adalah proses transformasi data dengan menggunakan metode *binning*. Metode *binning* adalah metode yang digunakan untuk mengelompokkan data beratribut numerik menjadi beberapa bin yang akan memudahkan dalam memahami persebaran data. Berdasarkan hasil analisis, ditemukan dua atribut yang bertipe data numerik dengan persebaran data yang cukup luas, yaitu los dan umur. Oleh karena itu proses binning akan dilakukan.

Untuk atribut umur akan dibagi menjadi 5 kategori, sesuai dengan standar usia WHO

Bin 1: umur <=1,
Bin 2: 2<= umur >=10,
Bin 3: 11<= umur >=19,
Bin 4: 20<= umur >=60,
Bin 5: umur > 60

In [21]:
```
# Binning dataset attribute umur

import numpy as np
import math
from sklearn.datasets import load_iris
from sklearn import datasets, linear_model, metrics

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

*Gambar 19*

Untuk fitur los yang bekaitan dengan jnpspsleep yaitu rawat inap atau rawat jalan. Kategori ini akan dibagi menjadi 4 bagian yaitu

0 : rawat jalan,
1-5 : short stay,
6- 10 : medium stay,
> 10 : long stay

In [22]:
```
# binning dataset attribute los
import numpy as np
import math
from sklearn.datasets import load_iris
from sklearn import datasets, linear_model, metrics

bin_limit = [-1, 1, 6, 11, 800]
category = ['outpatient', 'short stay', 'medium stay', 'long stay']
data['los'] = pd.cut(data['los'], bins=bin_limit, labels=category)
data
```

*Gambar 20*

### 3.5 Standardization

Pada tahapan standarisasi ini diperlukan agar dataset mendapatkan hasil yang lebih baik. Terlebih dahulu data harus disimpan ke dalam variabel X dan y seperti dibawah ini.

```
In [29]: # data.drop(['label'], axis=1, inplace=True)
         X = bpjs_final_data.iloc[:,:-1].values
         y = bpjs_final_data.iloc[:, -1].values
         y

Out[29]: array([1, 1, 1, ..., 0, 0, 0], dtype=int64)
```

*Gambar 21*

Penerapan standarisasi akan mengubah data mentah menjadi informasi yang dapat digunakan sebelum dilakukannya pemodelan. Teknik yang digunakan dengan menskalakan data sehingga memiliki mean = 0 dan standar deviasi =1. Standarisasi dilakukan dengan menggunakan fungsi standardscaler dan diperoleh hasil standarisasi sebagai berikut.

```
In [30]: # standardization
         from numpy import asarray
         from sklearn.preprocessing import StandardScaler


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

         print(X)
         # transform data
```

```
[[-0.21087184 -0.4873673  -0.51753857 ... -0.07424171 -0.65076355
  -0.09649292]
 [-0.21087184 -0.4873673   1.93222313 ... -0.07424171  2.42227887
  -0.09649292]
 [-0.21087184  2.05184056 -0.51753857 ... -0.07424171 -0.65076355
  -0.09649292]
 ...
 [-0.21087184 -0.4873673  -0.51753857 ... -0.07424171 -0.65076355
  -0.09649292]
 [-0.21087184  2.05184056 -0.51753857 ... -0.07424171  0.11749706
  -0.09649292]
 [-0.21087184 -0.4873673  -0.51753857 ... -0.07424171 -0.65076355
  -0.09649292]]
```

*Gambar 22*

# MODELLING

Tahapan selanjutnya pada metodologi CRISP-DM untuk melakukan binary classification dalam mendeteksi fraud pada BPJS 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

Dalam melakukan proses *data mining*, sebelum melakukan model diperlukan adanya perancangan bagaimana model yang akan dibangun. Analisis pengujian model adalah sebagai berikut.

1. Model dengan 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 klasifikasi menggunakan *Artificial Neural Network*. Hasil klasisifikasi yang dilakukan menghasilkan akurasi untuk data train dan data test masing-masing sebesar 0.64 dan 0.64

## 4.2 Build Modeling

*Binary Classification* dengan algoritma ANN akan dirancang menggunakan bahasa pemrograman *python* dengan menggunakan pustaka *python* yaitu *scikit-learn*. *Scikit-learn* adalah salah satu pustaka yang disediakan *python* untuk membangun model machine learning seperti klasifikasi ini. 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 ANN, tahap pertama yang dilakukan adalah membagi 3, yaitu: data latih, validasi, dan data uji dengan persentase 75% untuk data latih 15% untuk validasi, dan 10% untuk data uji. Data latih akan digunakan untuk membangun model dan data uji akan digunakan untuk menguji model yang telah dibangun.

In [33]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size=.10)
X_train, X_val, Y_train, Y_val = train_test_split(X_train, Y_train, test_size=.15)

print('X_train', X_train.shape)
print('X_val', X_val.shape)
print('X_test', X_test.shape)
```

```
X_train (153165, 120)
X_val (27030, 120)
X_test (20022, 120)
```

*Gambar 23*

Kemudian dilakukan pendefinisian informasi yang dibutuhkan dalam melakukan klasifikasi yaitu membuat model menggunakn algoritma yang sudah ditentukan.

```
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(64, ), activation='relu', max_iter=1000, epsilon=1e-08)
```

*Gambar 24*

Setelah dilakukan pemanggilan algoritma, maka tahapan selanjutnya adalah menampilkan akurasi data latih dan data uji.

```
from sklearn.metrics import accuracy_score

mlp.fit(X_train, Y_train)
val_predict = mlp.predict(X_val)
acc_val = accuracy_score(Y_val, val_predict)
print('Validation training accuration ANN:', acc_val)
```

```
Validation training accuration ANN: 0.6407325194228635
```

```
test_predict = mlp.predict(X_test)
acc_test = accuracy_score(Y_test, test_predict)
print('ANN testing accuracy:', acc_test)
```

```
ANN testing accuracy: 0.6422435321146739
```

*Gambar 25*

Selanjutnya kita menyimpan model yang sudah dihasilkan dengan menggunakan library pickle.

```
# save the model to disk
import nltk
import pickle
filename = 'model.pkl'
pickle.dump(MLPClassifier, open(filename, 'wb'))

# load the model from disk
loaded_model = pickle.load(open(filename, 'rb'))
# result = loaded_model.score(X_test, Y_test)
```

*Gambar 26*

# MODEL EVALUATION

Pada bab ini akan dijelaskan mengenai evaluasi terhadap model pendeteksi potensi kecurangan pada layanan BPJS yang dihasilkan menggunakan algoritma *Artificial Neural Network.* Evaluasi adalah fase interpretasi terhadap hasil penambangan data. Evaluasi akan dilakukan secara mendalam agar hasil pada tahap *modelling* sesuai dengan sasaran yang akan dicapai.

## 5.1 Evaluation of Modelling Result

Tahap ini dilakukan untuk mengetahui performa *binary classification* untuk mendeteksi *fraud* menggunakan *confusion matrix* berdasarkan dataset yang sudah digunakan yaitu data BPJS yang berasal dari kompetisi Hackathon. 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:

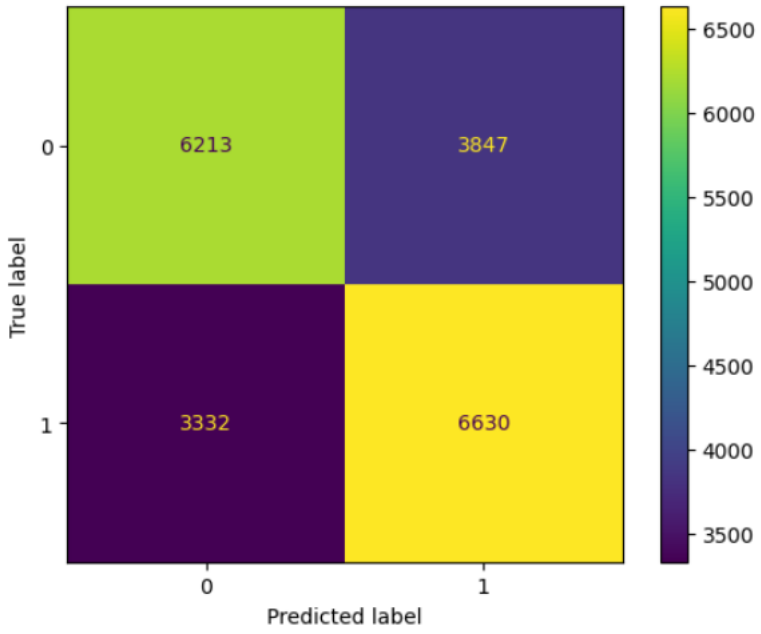
```
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score



# Calculating the precision score of classifier
print(f"Accuracy Score of the classifier is: {accuracy_score(Y_test, prediksi)}"
print(f"Precision Score of the classifier is: {precision_score(Y_test, prediksi)
print(f"Recall Score of the classifier is: {recall_score(Y_test, prediksi)}")
print(f"F1 Score of the classifier is: {f1_score(Y_test, prediksi)}")
```

```
Accuracy Score of the classifier is: 0.6414444111477375
Precision Score of the classifier is: 0.6328147370430467
Recall Score of the classifier is: 0.6655290102389079
F1 Score of the classifier is: 0.6487597240569499
```

*Gambar 27*

Berdasarkan hasil yang diperoleh dari pembangunan model dengan menggunakan algoritma ANN telah menghasilkan model dengan akurasi cukup baik dengan score > 0.60 dimana nilai ini sudah memenuhi standar dan ketentuan pembangunan proyek. Model yang dibangun telah cukup baik dalam menerapkan algoritma ANN untuk mendeteksi kecurangan pada layanan BPJS.

Selanjutnya evaluasi dilanjutkan dengan melakukan pemetaan kesesuaian output dari model menggunakan visualisasi heatmap, dan diperoleh hasil sebagai berikut:




*Gambar 28*

Karena penelitian ini merupakan *binary classification*, maka output akhir dari pemodelan ini adalah binary, dimana angka 0 memiliki arti terdapat tidak *fraud* dan 1 berarti terdapat *fraud*. Merujuk pada heatmap yang diperoleh dapat dilihat hubungan *predicted lable* dengan *true lable* dalam menentukan validasi data. Data valid merupakan data yang diprediksi tidak *fraud* berjumlah 6556 dan data yang diprediksi *fraud* 6303. Sementara untuk data yang diprediksi tidak *fraud* tetapi kebenarannya adalah *fraud* berjumlah 3493 dan data yang diprediksi *fraud* tetapi kebenarannya adalah tidak *fraud* berjumlah 3670.

## 5.2 Modelling Process Review

Tahap ini memeriksa Kembali semua tahapan yang dilakukan di awal dan berguna untuk memastikan bahwa tidak ada hal yang terabaikan atau terlewati. Dengan menggunakan metodologi CRISP-DM, maka dapat ditelaah bahwa:

- Tahapan EDA akan sangat membantu dalam pemilihan atribut yang berkaitan dalam mendeteksi terjadinya deteksi fraud pada layanan BPJS.
- Data preparation, pada proses pembersihan data dan transformasi data sehingga akan didapatkan data yang baik digunakan untuk modelling.

# DEPLOYMENT

Tahap terakhir pada pelaksanaan data mining menggunakan metodologi CRISP_DM adalah deployment. Dalam bab ini akan dijelaskan mengenai deployment yang akan dihasilkan.

# LAMPIRAN