

DOKUMEN PROYEK
12S4054 – PENAMBANGAN DATA
BPJS Case and Cost Prediction (Regression Problem)
using Decision Tree



Disusun oleh:

12S20009 Agnes Marpaung

12S20021 Sintia Lolita Silaen

12S20040 Esphi Aphelina Hutabarat

PROGRAM STUDI SARJANA SISTEM INFORMASI
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
INSTITUT TEKNOLOGI DEL
2023/204

DAFTAR ISI

DAFTAR ISI.....	2
BAB 1 BUSINESS UNDERSTANDING	6
1.1 Determine Business Objective	6
1.2 Determine Project Goal.....	7
1.3 Produce Project Plan	7
BAB 2 DATA UNDERSTANDING	9
2.1 Menelaah Data	9
2.2 Validation Data	12
BAB 3 DATA PREPARATION.....	17
3.1 Data Selection.....	17
3.2 Data Cleaning	18
3.3 Data Construct	19
BAB 4 MODELLING.....	20
4.1 Building Test Scenario	20
4.2 Build Model.....	21
BAB 5 EVALUATION	23
5.1 Evaluate Result	23
5.1.1 Evaluate result case prediction.....	24
5.1.2 Evaluate Result cost prediction	24

BAB 6 DEPLOYMENT	26
6.1 Plan Deployment	26

DAFTAR TABEL

Table 1. Jadwal Pelaksanaan Proyek	7
Table 2. Deskripsi Atribut.....	9
Table 3. Summary Hasil Evaluasi.....	25

DAFTAR GAMBAR

Gambar 1. Korelasi	16
Gambar 2. Halaman Case Prediction	31
Gambar 3. Cost Prediction	32
Gambar 4. Homepage	32

BAB 1

BUSINESS UNDERSTANDING

Business understanding merupakan langkah pertama dalam metodologi CRISP-DM untuk melakukan prediksi jumlah kasus dan unit cost pada sebuah wilayah yang diakibatkan dari penambahan rumah sakit. Pada bab ini akan menjelaskan terkait tahapan data mining untuk meningkatkan pemahaman dalam menentukan objektif bisnis, menentukan tujuan bisnis, dan membuat rencana proyek.

1.1 Determine Business Objective

Dalam kehidupan sehari-hari tentunya banyak pemanfaatan teknologi yang kita gunakan dan teknologi telah menjadi solusi utama dalam menyelesaikan berbagai masalah. Perkembangan teknologi yang pesat juga mempengaruhi berbagai sektor termasuk dalam pertumbuhan sektor kesehatan, khususnya dengan pertumbuhan jumlah rumah sakit di berbagai kota dan wilayah yang ada di Indonesia. Dengan pertumbuhan ini, menimbulkan tantangan dalam mengelola data dan menghadapi berbagai situasi yang kompleks. Maka dari itu pengerjaan proyek ini bertujuan untuk melakukan prediksi jumlah kasus dan unit cost di suatu wilayah tertentu dengan menggunakan data BPJS *Hackathon (Case and Cost Prediction)*.

Pada proyek ini, dilakukan pengembangan model data mining yang mampu melakukan prediksi jumlah kasus dan unit cost yang timbul akibat pertumbuhan jumlah rumah sakit di wilayah tertentu metode yang digunakan adalah Regression Problem dengan algoritma *Decision Tree Regression* yang merupakan salah satu teknik dari Data Mining untuk melakukan prediksi terhadap variabel kontinu atau nilai yang bersifat numerik yang terdiri dari variabel bebas(x) dan variabel tak bebas (y). Dalam konteks ini, algoritma *Decision Tree* akan mencoba memahami hubungan antara pertumbuhan rumah sakit dengan kenaikan jumlah kasus yang tercatat dan biaya yang dibutuhkan, memungkinkan analisis prediktif yang mendalam.

Data Mining merupakan konsep untuk mengeksplorasi informasi dari sekumpulan data. Proses ini menggunakan matematika, statistik, dan machine learning untuk mengekstrak pengetahuan. Data BPJS *Hackathon* merupakan data kategorikal yang memerlukan analisis sebelum membentuk model, dengan tujuan meningkatkan performa data dan mengidentifikasi faktor-faktor yang berpotensi menyebabkan *fraud* atau *non fraud*. Proyek ini memerlukan analisis data untuk mencapai prediksi jumlah kasus dan biaya, dimana analisis dilakukan sebelum data

dijadikan dasar pembentukan model. Model tersebut akan diimplementasikan menggunakan metode regresi dengan algoritma *Decision Tree* yang telah dipilih sebelumnya.

1.2 Determine Project Goal

Pada *Case and Cost Prediction* bertujuan untuk mengembangkan model data mining yang mampu melakukan prediksi jumlah kasus dan unit cost di sebuah daerah sebagai hasil dari penambahan rumah sakit berdasarkan data BPJS *Hackathon*.

1.3 Produce Project Plan

Tahap perencanaan yang dilakukan untuk mencapai tujuan pengerjaan adalah sebagai berikut:

Table 1. Jadwal Pelaksanaan Proyek

Tahapan	Waktu Pengerjaan	Kegiatan
<i>Business Understanding</i>	3 hari	Menentukan objektif bisnis, menentukan tujuan bisnis, dan membuat rencana proyek.
<i>Data Understanding</i>	4 hari	Mengumpulkan data, menelaah data, memvalidasi data
<i>Data Preparation</i>	3 hari	Memilih data, membersihkan data, mengkonstruksi data, menentukan label data, dan mengintegrasikan data.
<i>Modeling</i>	3 hari	Membangun skenario pengujian dan membangun model.
<i>Evaluation</i>	3 hari	Melakukan evaluasi hasil pemodelan dan melakukan <i>review</i> terhadap proses pemodelan.
<i>Deployment</i>	4 hari	Membuat rencana <i>deployment</i> model, <i>Monitoring and Maintenance</i> rencana

		<i>deployment</i> model dan meninjau proyek.
--	--	--

BAB 2

DATA UNDERSTANDING

Dalam tahapan data understanding yang merupakan tahapan pemahaman terhadap data yang akan digunakan, tahapan ini dimulai dari mendeskripsikan data dan memahami data yang akan digunakan dalam penelitian.

2.1 Menelaah Data

Dataset yang digunakan untuk memprediksi jumlah kasus dan biaya unit di suatu daerah akibat penambahan Rumah Sakit kerja sama adalah *case_cost_prediction_train.csv*. Dataset ini terdiri dari 57.971 observasi dan memiliki 36 variabel. Berikut adalah tabel yang membahas atribut dalam dataset.

Table 2. Deskripsi Atribut

No.	Atribut	Deskripsi
1	<i>row_id</i>	ID dari setiap data
2	<i>tgl_pelayanan</i>	periode bulan pelayanan di rumah sakit
3	<i>kddati2</i>	kode kabupaten/kota
4	<i>tkp</i>	tingkat pelayanan; 30:rawat jalan; 40:rawat inap
5	<i>peserta</i>	jumlah peserta akhir pada kabupaten/kota periode tersebut
6	<i>a,b,c,...,sd</i>	tipe rumah sakit yang melayani peserta JKN-KIS
7	<i>case</i>	jumlah kunjungan rumah sakit
8	<i>unit_cost</i>	jumlah biaya pelayanan rumah sakit

Berdasarkan dataset tersebut, akan dilakukan EDA (*Exploratory Data Analysis*) terhadap dataset *case_cost_prediction_train.csv* untuk menganalisis karakteristik utamanya. Dalam pengerjaan proyek ini, tidak semua atribut dari dataset digunakan, melainkan hanya beberapa atribut yang relevan terhadap tujuan penelitian. Oleh karena itu, atribut yang paling sesuai untuk memprediksi jumlah kasus dan biaya unit adalah atribut *case*, atribut *unit_cost*, dan beberapa atribut lain yang relevan. Berikut adalah beberapa hipotesis terkait atribut dalam dataset yang akan digunakan.

- atribut *kddati2* digunakan untuk mengetahui kasus per kabupaten/kota berdasarkan kode yang telah ditetapkan.
- atribut *peserta* digunakan untuk mengetahui jumlah peserta.
- atribut *case* digunakan dalam mengetahui kasus kunjungan ke rumah sakit.
- atribut *unit_cost* digunakan untuk mengetahui biaya pelayanan rumah sakit.

Berdasarkan hipotesis-hipotesis tersebut, atribut *kddati2*, *peserta*, *case*, dan *unit_cost* memiliki pengaruh signifikan terhadap jumlah kasus dan biaya unit. Atribut-atribut ini dianggap relevan dan dapat digunakan dalam pengembangan model data mining untuk meramalkan jumlah kasus dan biaya unit di suatu daerah sebagai dampak dari penambahan Rumah Sakit kerja sama sesuai dengan tujuan proyek.

- `data.info()`

Untuk memberikan informasi ringkas mengenai dataset mengenai jumlah baris dan column, nama column, tipe data, dan jumlah nilai non-null.

```
RangeIndex: 57971 entries, 0 to 57970
Data columns (total 36 columns):
#   Column                Non-Null Count  Dtype
---  -
0   row_id                57971 non-null  int64
1   tglpelayanan          57971 non-null  object
2   kddati2               57971 non-null  int64
3   tkp                   57971 non-null  int64
4   peserta               57971 non-null  int64
5   a                     57971 non-null  int64
6   b                     57971 non-null  int64
7   c                     57971 non-null  int64
8   cb                    57971 non-null  int64
9   d                     57971 non-null  int64
10  ds                    57971 non-null  int64
11  gd                    57971 non-null  int64
12  hd                    57971 non-null  int64
13  i1                    57971 non-null  int64
14  i2                    57971 non-null  int64
15  i3                    57971 non-null  int64
16  i4                    57971 non-null  int64
17  kb                    57971 non-null  int64
18  kc                    57971 non-null  int64
19  kg                    57971 non-null  int64
20  ki                    57971 non-null  int64
21  kj                    57971 non-null  int64
22  kk                    57971 non-null  int64
23  kl                    57971 non-null  int64
24  km                    57971 non-null  int64
25  ko                    57971 non-null  int64
26  kp                    57971 non-null  int64
27  kt                    57971 non-null  int64
28  ku                    57971 non-null  int64
29  s                     57971 non-null  int64
30  sa                    57971 non-null  int64
31  sb                    57971 non-null  int64
32  sc                    57971 non-null  int64
33  sd                    57971 non-null  int64
34  case                  57971 non-null  int64
35  unit_cost             57971 non-null  float64
```

- data.head (10)

	row_id	tglpelayanan	kddati2	tkp	peserta	a	b	c	cb	d	...	kp	kt	ku	s	sa	sb	sc	sd	case	unit_cost
0	1	2014-04-01 00:00:00	332	40	179530	0	0	1	0	1	...	0	0	0	0	0	0	0	0	266	3.597440e+06
1	2	2016-11-01 00:00:00	54	40	104782	0	1	0	0	0	...	0	0	1	0	0	0	1	1	2453	4.951008e+06
2	3	2016-05-01 00:00:00	323	30	280645	0	0	1	0	0	...	0	0	0	0	0	0	0	0	1690	1.984208e+05
3	4	2018-11-01 00:00:00	318	40	178685	0	1	0	0	0	...	0	0	0	0	0	0	0	3	1321	4.008756e+06
4	5	2019-10-01 00:00:00	150	30	1199321	1	0	1	0	0	...	1	0	0	0	0	1	0	5	73056	3.072272e+05
5	6	2014-10-01 00:00:00	37	40	110377	0	1	0	0	0	...	0	0	0	0	0	0	1	0	1074	3.426614e+06
6	7	2020-05-01 00:00:00	379	40	105699	0	0	1	0	0	...	0	0	0	0	0	0	0	0	262	2.943943e+06
7	8	2015-09-01 00:00:00	110	40	152065	0	0	1	0	0	...	0	0	0	0	0	0	1	1	1102	3.971695e+06
8	9	2015-03-01 00:00:00	303	30	179081	0	1	0	0	1	...	0	0	0	0	0	0	0	0	5533	2.317746e+05
9	10	2017-11-01 00:00:00	49	30	227227	0	0	1	0	0	...	0	0	0	0	0	0	0	0	6426	1.903432e+05

- data.dtypes

```

row_id          int64
tglpelayanan    object
kddati2         int64
tkp             int64
peserta         int64
a              int64
b              int64
c              int64
cb             int64
d              int64
ds             int64
gd             int64
hd             int64
i1             int64
i2             int64
i3             int64
i4             int64
kb             int64
kc             int64
kg             int64
ki             int64
kj             int64
kk             int64
kl             int64
km             int64
ko             int64
kp             int64
kt             int64
ku             int64
s              int64
sa             int64
sb             int64
sc             int64
sd             int64
case            int64
unit_cost       float64
dtype: object

```

- `data.tail()`

```
[17]: #Menghitung dan menampilkan beberapa data tail
data.tail()
```

```
[17]:
```

	row_id	tglpelayanan	kddati2	tkp	peserta	a	b	c	cb	d	...	kp	kt	ku	s	sa	sb	sc	sd	case	unit_cost	
57966	57967	2019-03-01 00:00:00		241	40	157213	0	0	1	0	0	...	0	0	0	0	0	0	0	0	410	3.443332e+06
57967	57968	2019-09-01 00:00:00		338	30	402173	0	0	1	0	0	...	0	0	0	0	0	0	0	0	8272	2.128621e+05
57968	57969	2016-06-01 00:00:00		241	40	99401	0	0	1	0	0	...	0	0	0	0	0	0	0	0	294	3.028611e+06
57969	57970	2018-01-01 00:00:00		147	40	509495	0	0	1	0	0	...	0	0	0	0	0	0	2	0	1983	3.629365e+06
57970	57971	2016-09-01 00:00:00		204	30	770169	0	1	1	0	0	...	0	0	0	0	0	1	2	1	16679	2.956852e+05

5 rows × 36 columns

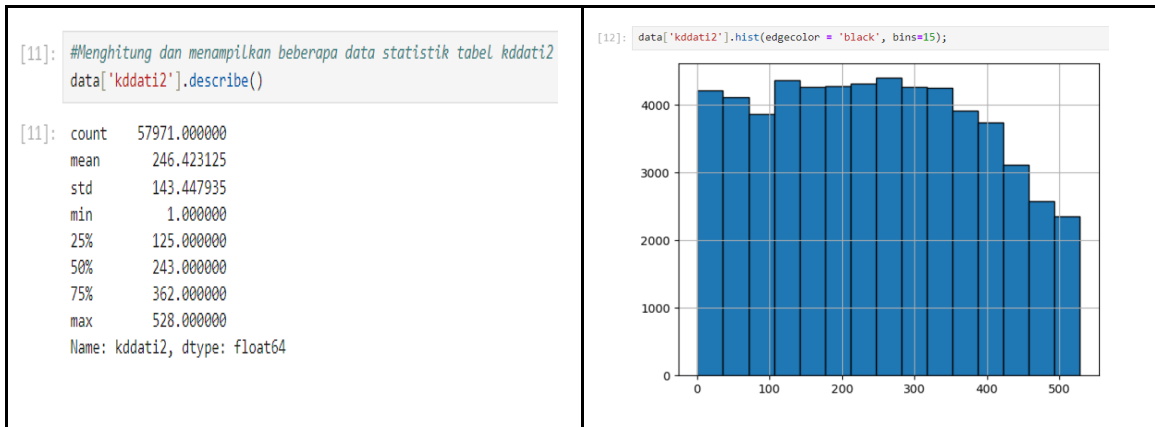
- `data.describe()`

	row_id	kddati2	tkp	peserta	a	b	c	cb	d	ds	...
count	57971.000000	57971.000000	57971.000000	5.797100e+04	57971.000000	57971.000000	57971.000000	57971.000000	57971.000000	57971.0	...
mean	28986.000000	246.423125	34.990081	3.562209e+05	0.041538	0.388574	0.788894	0.000380	0.281710	0.0	...
std	16734.930565	143.447935	5.000033	4.120323e+05	0.210390	0.660382	0.679786	0.019477	0.595284	0.0	...
min	1.000000	1.000000	30.000000	8.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	...
25%	14493.500000	125.000000	30.000000	1.127735e+05	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	...
50%	28986.000000	243.000000	30.000000	1.975800e+05	0.000000	0.000000	1.000000	0.000000	0.000000	0.0	...
75%	43478.500000	362.000000	40.000000	4.386935e+05	0.000000	1.000000	1.000000	0.000000	0.000000	0.0	...
max	57971.000000	528.000000	40.000000	3.328509e+06	2.000000	8.000000	6.000000	1.000000	5.000000	0.0	...

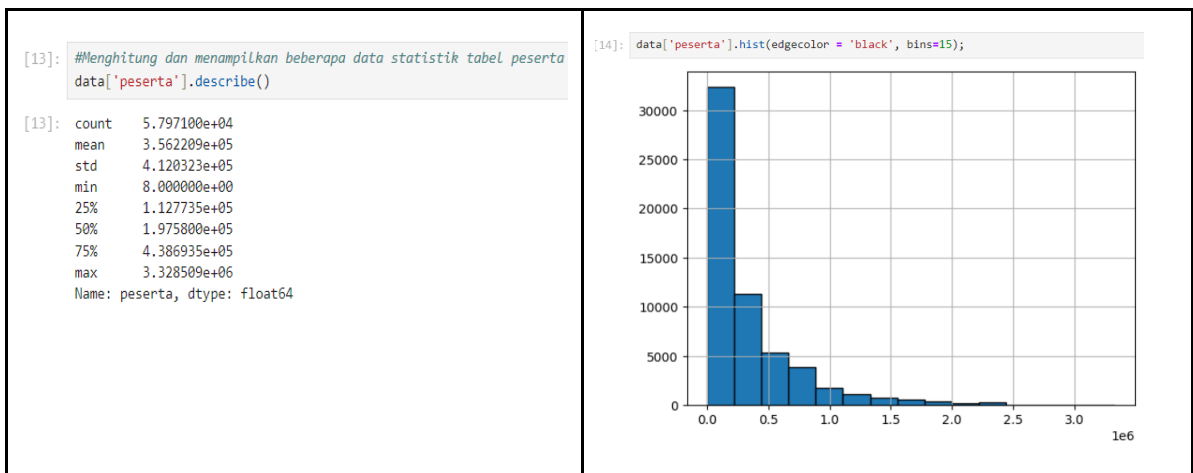
2.2 Validation Data

Pada sub bab ini akan dijelaskan tahap validasi data untuk melakukan pemeriksaan menyeluruh terhadap keakuratan, kelengkapan, dan kualitas sumber data sebelum digunakan untuk analisis atau pengolahan lebih lanjut. Tahapan ini melibatkan identifikasi dan penanganan potensi kesalahan seperti nilai yang hilang (*missing value*) atau *noise* dalam data. Pengecekan memungkinkan untuk membersihkan dan menormalkan data agar menjadi konsisten, lengkap, dan akurat, memastikan bahwa informasi yang diolah dapat diandalkan dalam pengambilan keputusan dan analisis mendatang.

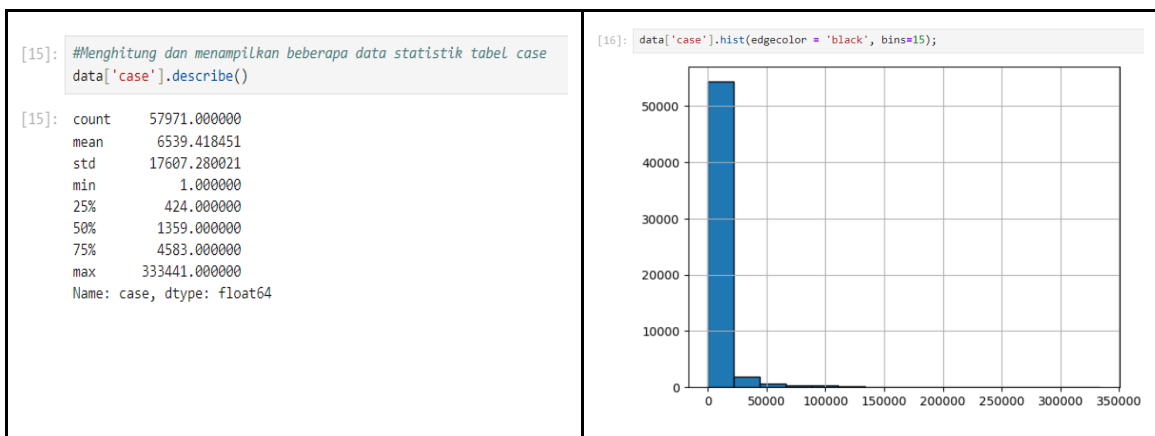
- Atribut *kddati2*



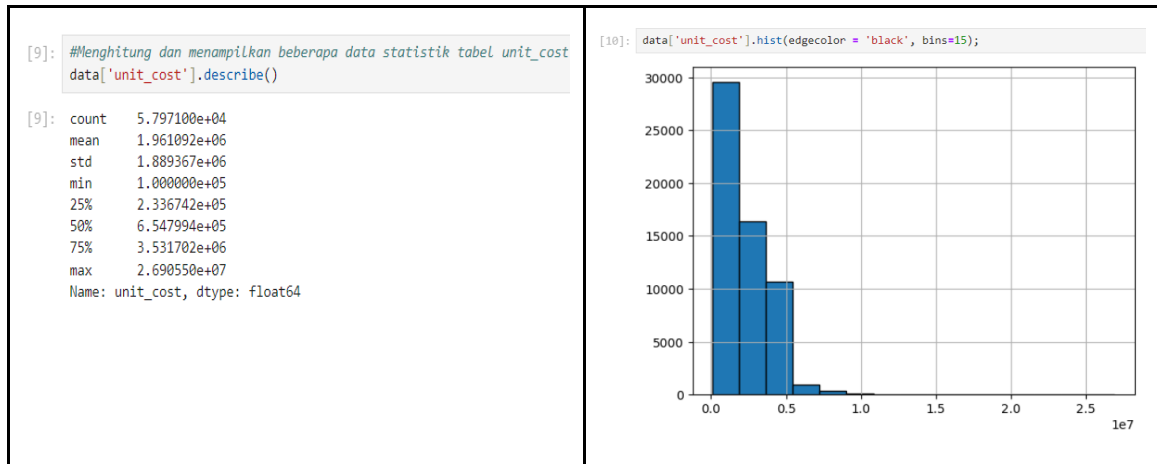
- Atribut *peserta*



- Atribut *case*



- Atribut *unit_cost*

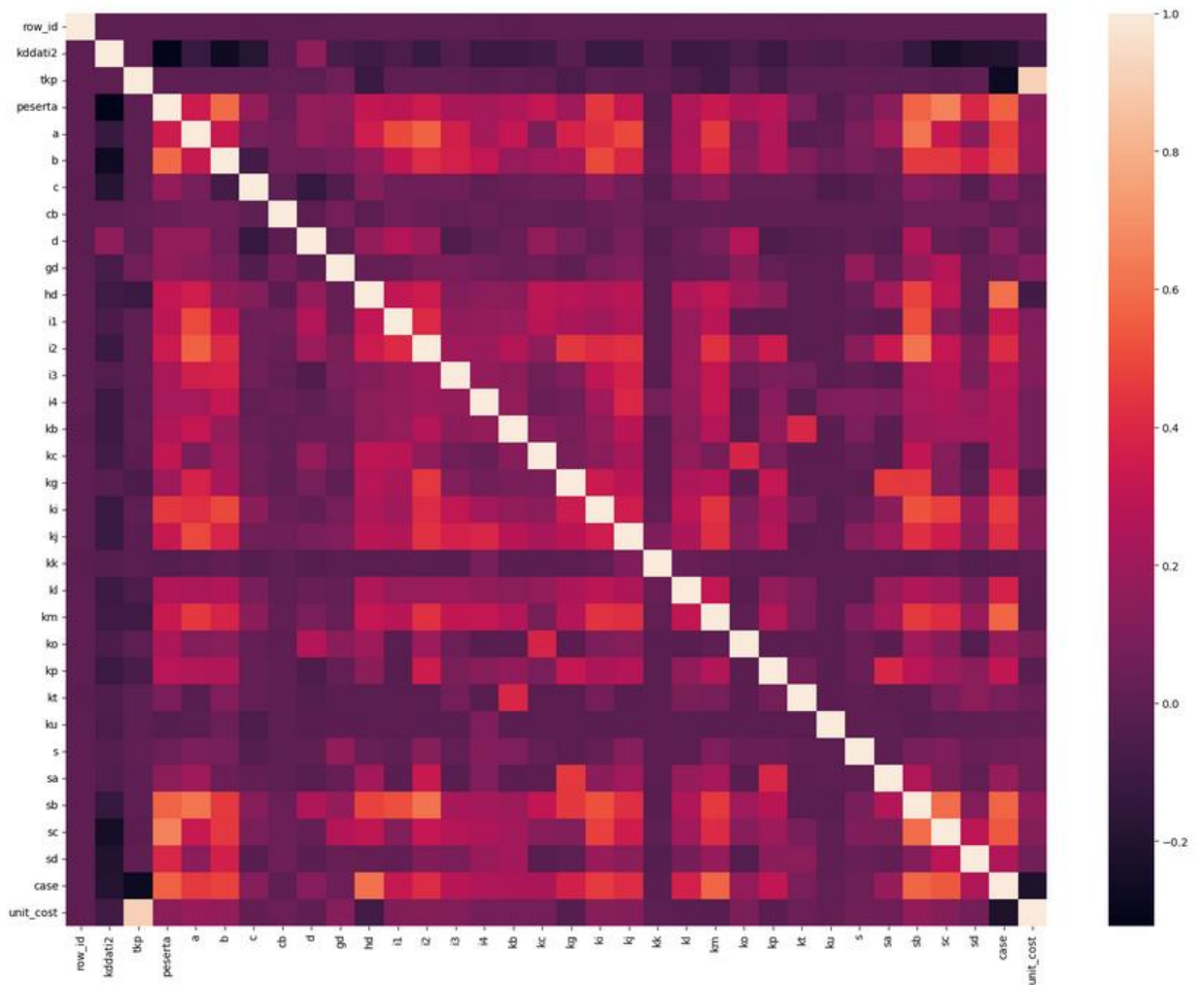


- Memeriksa Missing Value

```
row_id      0
tglpelayan  0
kddati2     0
tkp         0
peserta     0
a           0
b           0
c           0
cb          0
d           0
ds          0
gd          0
hd          0
i1          0
i2          0
i3          0
i4          0
kb          0
kc          0
kg          0
ki          0
kj          0
kk          0
kl          0
km          0
ko          0
kp          0
kt          0
ku          0
s           0
sa          0
sb          0
sc          0
sd          0
case        0
unit_cost   0
dtype: int64
```

- Memeriksa Corellation

	row_id	kddati2	tkp	peserta	a	b	c	cb	d	gd	...	kp	kt	ku
row_id	1.000000	0.000232	0.004808	0.001004	-0.000340	0.003964	-0.004531	-0.002933	-0.002808	-0.000611	...	-0.000030	-0.006730	-0.002346
kddati2	0.000232	1.000000	0.000889	-0.322811	-0.123548	-0.269241	-0.185139	-0.004898	0.152947	-0.065246	...	-0.115667	-0.045977	-0.057413
tkp	0.004808	0.000889	1.000000	-0.000516	-0.001576	-0.002176	0.001160	0.015981	-0.000017	0.062540	...	-0.067691	0.000394	-0.001530
peserta	0.001004	-0.322811	-0.000516	1.000000	0.347552	0.590898	0.170167	0.034477	0.166506	0.157286	...	0.279015	0.089745	-0.026350
a	-0.000340	-0.123548	-0.001576	0.347552	1.000000	0.327321	0.075425	0.055088	0.163857	0.127321	...	0.259888	-0.013020	-0.008450
b	0.003964	-0.269241	-0.002176	0.590898	0.327321	1.000000	-0.081680	0.058274	0.045645	0.083629	...	0.255288	0.111581	0.039628
c	-0.004531	-0.185139	0.001160	0.170167	0.075425	-0.081680	1.000000	-0.000463	-0.135705	-0.038147	...	0.023226	0.021252	-0.049670
cb	-0.002933	-0.004898	0.015981	0.034477	0.055088	0.058274	-0.000463	1.000000	-0.003270	0.070216	...	0.004731	-0.001285	-0.000834
d	-0.002808	0.152947	-0.000017	0.166506	0.163857	0.045645	-0.135705	-0.003270	1.000000	-0.007870	...	-0.050565	-0.031207	-0.020255
gd	-0.000611	-0.065246	0.062540	0.157286	0.127321	0.083629	-0.038147	0.070216	-0.007870	1.000000	...	0.018375	-0.004467	-0.006021
hd	0.003669	-0.096964	-0.116800	0.310015	0.351359	0.162438	0.118948	-0.002511	0.169492	0.026629	...	0.134070	-0.008500	-0.005517
i1	0.005091	-0.058595	0.005001	0.295560	0.494127	0.310388	0.052984	0.054990	0.263420	0.030562	...	-0.014971	-0.005198	-0.003374
i2	0.002629	-0.119370	-0.000147	0.341266	0.565330	0.410832	0.052202	0.033459	0.190907	0.094811	...	0.347906	-0.012628	-0.008196
i3	-0.001426	-0.037825	0.004874	0.237976	0.363718	0.371573	0.062879	0.017370	-0.047309	0.087277	...	0.092460	0.068279	-0.012372
i4	0.004480	-0.114911	-0.002619	0.223160	0.222083	0.316869	0.010559	0.030110	-0.001370	0.058824	...	0.129149	-0.021867	0.103629
kb	-0.004746	-0.110705	-0.001092	0.248437	0.317914	0.177552	0.035230	0.008066	0.031368	0.036163	...	0.162028	0.390530	-0.006279
kc	0.006690	-0.085424	-0.006171	0.310727	0.090012	0.206765	0.048980	0.017380	0.164132	0.045074	...	0.089976	-0.006138	-0.003984
kg	0.006660	-0.022723	-0.054652	0.194227	0.377772	0.234994	0.045895	0.005512	0.078023	0.003314	...	0.318410	-0.005092	-0.003305
ki	0.002591	-0.118277	-0.000769	0.451755	0.422128	0.498765	0.141477	0.036418	0.011262	0.075827	...	0.240699	0.074573	-0.012844
kj	-0.000866	-0.117813	-0.009578	0.324682	0.496866	0.387548	0.062993	0.044783	0.082675	0.112698	...	0.264164	-0.017849	-0.011585
kk	-0.006017	-0.025941	-0.003396	-0.029748	-0.011675	0.017070	-0.026853	-0.001152	-0.027984	-0.008319	...	-0.011230	-0.003899	-0.002531
kl	-0.000441	-0.110963	-0.050570	0.249870	0.239116	0.251806	0.083190	0.002463	0.030514	0.018393	...	0.159132	0.093030	-0.012590
km	0.005440	-0.100744	-0.100317	0.331195	0.452170	0.382076	0.140418	0.011333	0.087791	0.028728	...	0.259560	0.075224	-0.008848
ko	0.000165	-0.058261	-0.001250	0.240889	0.115664	0.133682	0.020771	-0.001227	0.262164	0.137131	...	-0.011960	-0.004153	-0.002695
kp	-0.000030	-0.115667	-0.067691	0.279015	0.259888	0.255288	0.023226	0.004731	-0.050565	0.018375	...	1.000000	0.068761	-0.008128
kt	-0.006730	-0.045977	0.000394	0.089745	-0.013020	0.111581	0.021252	-0.001285	-0.031207	-0.004467	...	0.068761	1.000000	-0.002822
ku	-0.002346	-0.057413	-0.001530	-0.026350	-0.008450	0.039628	-0.049670	-0.000834	-0.020255	-0.006021	...	-0.008128	-0.002822	1.000000
s	0.003559	-0.027376	0.012026	0.039948	0.086596	0.077668	-0.025408	-0.000974	0.004102	0.160220	...	0.035320	0.006263	-0.002140
sa	0.005743	-0.034931	-0.001064	0.145526	0.203044	0.041055	0.013400	-0.000876	-0.021282	0.026509	...	0.393952	-0.002966	-0.001925
sb	0.005183	-0.135862	-0.001505	0.569431	0.618986	0.453759	0.127382	0.039216	0.251740	0.170699	...	0.282125	-0.010320	-0.010568
sc	0.006314	-0.245343	-0.002684	0.654574	0.329413	0.451656	0.086505	0.041790	0.027522	0.263922	...	0.198363	0.084365	0.000903
sd	0.000602	-0.205379	0.004531	0.392445	0.140970	0.366047	-0.013533	0.047226	-0.011800	0.031431	...	0.144049	0.137966	0.007078
case	0.007574	-0.195391	-0.279366	0.561711	0.455569	0.479891	0.125083	0.004755	0.127811	0.051567	...	0.304688	0.085972	0.012336
unit_cost	0.004244	-0.093774	0.907384	0.141987	0.180618	0.166028	0.015957	0.039635	0.003352	0.125652	...	-0.008068	0.040727	0.014184



Gambar 1. Korelasi

BAB 3

DATA PREPARATION

Data preparation adalah tahap dalam proses analisis data yang dilakukan untuk menghasilkan data yang berkualitas baik. Tahapan dalam data preparation meliputi data *Selection*, data *Cleaning*, data *Construct*, dan Labeling Data. Data preparation dilakukan setelah pengumpulan data awal pada fase business understanding dan sebelum analisis data dilakukan. Tahapan data preparation meliputi beberapa proses, seperti ekstraksi, transformasi, pembersihan, standarisasi, integrasi, validasi, *formatting*, dan *summarization*.

3.1 Data Selection

Tahapan ini melibatkan proses pemilihan atribut atau kolom yang relevan dalam sebuah dataset. Atribut yang tidak diperlukan akan dihapus. Atribut yang tidak digunakan diidentifikasi dan dimasukkan ke dalam variabel bernama 'to_drop', kemudian dilakukan penghapusan kolom sesuai dengan daftar atribut yang terdapat dalam 'to_drop'.

```
[22]: from sklearn import preprocessing
      lab_enc = preprocessing.LabelEncoder()
```

```
[23]: data
```

```
[23]:
```

	row_id	tglpelayanan	kddat12	tkp	peserta	a	b	c	cb	d	...	kp	kt	ku	s	sa	sb	sc	sd	case	unit_cost
0	1	2014-04-01 00:00:00	332	40	179530	0	0	1	0	1	...	0	0	0	0	0	0	0	0	266	3.597440e+06
1	2	2016-11-01 00:00:00	54	40	104782	0	1	0	0	0	...	0	0	1	0	0	0	1	1	2453	4.951008e+06
2	3	2016-05-01 00:00:00	323	30	280645	0	0	1	0	0	...	0	0	0	0	0	0	0	0	1690	1.984208e+05
3	4	2018-11-01 00:00:00	318	40	178685	0	1	0	0	0	...	0	0	0	0	0	0	0	3	1321	4.008756e+06
4	5	2019-10-01 00:00:00	150	30	1199321	1	0	1	0	0	...	1	0	0	0	0	1	0	5	73056	3.072272e+05
...
57966	57967	2019-03-01 00:00:00	241	40	157213	0	0	1	0	0	...	0	0	0	0	0	0	0	0	410	3.443332e+06
57967	57968	2019-09-01 00:00:00	338	30	402173	0	0	1	0	0	...	0	0	0	0	0	0	0	0	8272	2.128621e+05
57968	57969	2016-06-01 00:00:00	241	40	99401	0	0	1	0	0	...	0	0	0	0	0	0	0	0	294	3.028611e+06
57969	57970	2018-01-01 00:00:00	147	40	509495	0	0	1	0	0	...	0	0	0	0	0	0	2	0	1983	3.629365e+06
57970	57971	2016-09-01 00:00:00	204	30	770169	0	1	1	0	0	...	0	0	0	0	0	1	2	1	16679	2.956852e+05

57971 rows x 36 columns

```
[24]: to_drop = ['tglpelayanan',
               'tkp',
               'row_id']
       data.drop(to_drop, inplace = True, axis = 1)
```

```
[25]: data.head()
```

```
[25]:
```

	kddati2	peserta	a	b	c	cb	d	ds	gd	hd	...	kp	kt	ku	s	sa	sb	sc	sd	case	unit_cost
0	332	179530	0	0	1	0	1	0	0	0	...	0	0	0	0	0	0	0	0	266	3.597440e+06
1	54	104782	0	1	0	0	0	0	0	0	...	0	0	1	0	0	0	1	1	2453	4.951008e+06
2	323	280645	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	1690	1.984208e+05
3	318	178685	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	3	1321	4.008756e+06
4	150	1199321	1	0	1	0	0	0	0	0	...	1	0	0	0	0	1	0	5	73056	3.072272e+05

5 rows × 33 columns

3.2 Data Cleaning

Data cleaning adalah proses dalam menganalisis data yang bertujuan untuk mengidentifikasi, mengoreksi, dan menghapus kesalahan atau ketidaksesuaian dalam dataset. Data cleaning sangat penting dalam proses data mining karena dapat meningkatkan kualitas data, meningkatkan akurasi, dan meningkatkan keandalan hasil analisis. Tahapan dalam data cleaning meliputi mendeteksi kesalahan atau data yang rusak, memperbaiki atau menghapus data yang tidak diperlukan atau yang duplikat, dan memastikan konsistensi dan keakuratan data.

```
[30]: data.isnull().sum()
```

```
[30]: kddati2      0
       peserta     0
       case       0
       unit_cost   0
       tipe_gabungan 0
       dtype: int64
```

```
[31]: data.dtypes
```

```
[31]: kddati2      int64
       peserta     int64
       case       int64
       unit_cost   float64
       tipe_gabungan int64
       dtype: object
```

3.3 Data Construct

Data yang tersedia memiliki nilai boolean yang menunjukkan bahwa data tersebut tidak memiliki nilai kosong atau missing value, sehingga tidak perlu menggunakan fungsi `dropna()` untuk menghapusnya. Selanjutnya, perlu dilakukan konstruksi data sebelum proses pemodelan. Konstruksi data adalah bagian dari transformasi data yang mencakup representasi fitur, penentuan korelasi, dan integrasi data. Representasi fitur digunakan untuk mengurangi kompleksitas data, meningkatkan akurasi prediksi, dan memilih fitur yang paling berpengaruh.

Untuk memprediksi biaya, digunakan atribut 'kddati2', 'peserta', 'tkp', dan 'tipe_gabungan'. Sedangkan untuk memprediksi kasus, digunakan atribut 'kddati2', 'peserta', 'unit_cost', 'tipe_gabungan', dan 'tkp'. 'Tipe Gabungan' adalah jumlah dari semua jenis rumah sakit yang melayani peserta JKN-KIS.

BAB 4

MODELLING

Pada bab sebelumnya, persiapan data telah dilakukan untuk mempersiapkan model yang akan dikembangkan. Selanjutnya pada bab ini akan mengeksplorasi pembangunan model Regresi Prediksi dengan menerapkan algoritma *Decision Tree Regressor*. Adapun tujuan dibangunnya model ini pada konteks BPJS Hackathon adalah untuk memprediksi jumlah kasus dan biaya unit di suatu daerah sebagai hasil dari penambahan Rumah Sakit pada wilayah tersebut. Selanjutnya regresi mempunyai tujuan untuk melakukan proses prediksi nilai yang bersifat kontinu, dan ada berbagai macam algoritma yang dapat digunakan untuk prediksi kasus ini, salah satunya adalah *Long Short-Term Memory*. Adapun alasan kelompok kami memilih untuk menggunakan Decision Tree Regression yaitu karena kemampuannya dalam mengingat informasi dalam jangka waktu yang panjang sambil menghapus informasi yang sudah tidak relevan. Harapan kami menggunakan algoritma ini supaya lebih memudahkan pemahaman dan interpretasi kinerjanya dalam menyelesaikan proyek Data Mining ini. Oleh karena itu, tahap-tahap yang akan dilakukan dalam pembuatan model ini disajikan sebagai berikut.

4.1 Building Test Scenario

Dalam membangun *test scenario* untuk prediksi *case and cost (Regression Problem)* menggunakan *Decision Tree*. Data yang ditambahkan dalam melakukan pemodelan yaitu pemilihan variabel yang relevan seperti *tglpelayanan*, *kddati2* (kode wilayah), *peserta*, *case*, dan *unit_cost* yang kemudian dibagi menjadi data pelatihan dan uji. Selanjutnya, dengan menerapkan algoritma *Decision Tree*, model yang telah dilatih bertujuan untuk memahami hubungan antara variabel input dan output, yaitu jumlah kasus dan biaya pelayanan rumah sakit. Proses evaluasi dilakukan dengan menggunakan data uji untuk mengukur seberapa akurat model *Decision Tree* dalam memprediksi kasus dan biaya, memberikan gambaran yang lebih baik dalam perencanaan dan pengelolaan rumah sakit dalam mengantisipasi kasus dan estimasi biaya pelayanan.

4.2 Build Model

Dalam proses pengembangan model, dataset yang telah disiapkan melalui tahap *preprocessing*, seperti yang telah dibahas pada bab sebelumnya. Adapun *library* yang diperlukan dalam pembuatan model ini adalah sebagai berikut. Pertama sekali, digunakan *library* 'sklearn' untuk membangun model *Decision Tree Regression*. Kemudian model ini akan diuji pada dataset BPJS yang telah disiapkan. *Library* ini memungkinkan implementasi model regresi pohon keputusan serta proses pelatihan dan pengujian model pada data yang telah disiapkan. Dibawah ini adalah proses dalam pembangunan model yang bertujuan untuk mencari nilai dari *case and cost*.

- Dibutuhkan *library* sklearn untuk membangun model *Decision Tree Regression*. Selanjutnya, model akan dijadikan sebagai bahan latihan dan pengujian menggunakan dataset dari BPJS.

```
import numpy as np
import pandas as pd
from sklearn import metrics
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.compose import ColumnTransformer
```

- Selanjutnya mendefinisikan fitur yang digunakan untuk melakukan prediksi.

Berikut merupakan fitur yang digunakan dalam kasus '*case*'

```
X = data[['kddati2', 'peserta', 'unit_cost', 'tkp', 'tipe_gabungan']]
y = data['case']
```

Berikut merupakan fitur yang digunakan dalam kasus '*cost*'

```
x = data[['kddati2', 'peserta', 'tkp', 'tipe_gabungan']]
Y = data['unit_cost']
```

- Setelah fitur didefinisikan, langkah selanjutnya adalah membagi data menjadi data training dan data test.

Untuk Cost Prediction menggunakan rasio 90:10

```
X = data.drop('case', axis=1)
y = data['case']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
```

```
X = data[['kddati2', 'peserta', 'unit_cost', 'tkp', 'tipe_gabungan']]
y = data['case']
```

```
print("Banyak data latih setelah train validation split:", len(X_train))
print("Banyak data uji setelah train validation split:", len(X_test))
```

```
Banyak data latih setelah train validation split: 46376
Banyak data uji setelah train validation split: 11595
```

Untuk case menggunakan rasio 80:20

```
x_train, x_test, Y_train, Y_test = train_test_split(x,Y, test_size = 0.2)
```

```
print("Banyak data latih setelah train validation split:", len(X_train))
print("Banyak data uji setelah train validation split:", len(X_test))
```

```
Banyak data latih setelah train validation split: 52173
Banyak data uji setelah train validation split: 5798
```

- Setelah membagi data, langkah selanjutnya adalah menambahkan lapisan pada model Decision Tree Regression. Hal ini melibatkan inisialisasi model Sequential yang akan mengimplementasikan jaringan saraf Decision Tree Regression. Proses ini termasuk menambahkan lapisan normalisasi batch dan lapisan keluaran padat. Setelah itu, akan dicetak untuk mendapatkan gambaran model yang akan dibuat. Dengan demikian, kita akan membuat model regresi Decision Tree-nya.

```
# case prediction
from sklearn.tree import DecisionTreeRegressor
dtrcase = DecisionTreeRegressor(random_state=42)
dtrcase.fit(X_train, y_train)
```

- Selanjutnya mendefinisikan objek y_pred dengan untuk memprediksi hasil model regresi yang akan dibangun.

```
y_pred = dtrcase.predict(X_test)
```

BAB 5

EVALUATION

Pada bab *Evaluation* akan dijelaskan tentang bagaimana evaluasi yang diperoleh terhadap model *Decision tree* dalam memprediksi case and cost dalam sebuah daerah akibat penambahan Rumah Sakit yang terjadi dalam suatu wilayah. Evaluasi ini dilakukan dengan tujuan agar dapat mengetahui bagaimana hasil pada tahap modelling terhadap tujuan yang ingin dicapai seperti yang telah dijelaskan dalam business understanding.

5.1 Evaluate Result

Evaluasi terhadap model dalam memprediksi Case and Cost dalam suatu wilayah sebagai dampak dari penambahan Rumah Sakit dapat dilaksanakan dengan berbagai metode, salah satunya adalah melalui penerapan algoritma Decision Tree Regression. Dalam menilai kinerja model yang menggunakan Decision Tree Regression dalam memprediksi Case and Cost, dapat dilakukan dengan menggunakan beberapa metrik evaluasi seperti mean absolute error (MAE). MAE digunakan untuk mengukur sejauh mana perbedaan antara nilai prediksi dan nilai sebenarnya. Selain itu, metrik evaluasi yang digunakan adalah Mean Absolute Percentage Error (MAPE), yang sering digunakan dalam analisis deret waktu. Kedua metrik ini memberikan informasi tentang tingkat kesalahan dalam prediksi model, berguna untuk mengevaluasi kemampuan model dalam memprediksi jumlah kasus dan biaya unit di suatu wilayah akibat penambahan Rumah Sakit. Oleh karena itu, Decision Tree menjadi pilihan yang tepat untuk memprediksi jumlah kasus dan biaya unit di wilayah tersebut.

5.1.1 Evaluate result case prediction

Pada bagian ini dilakukan evaluate result pada *case* prediction agar mengetahui bagaimana kinerja model yang dibangun.

```
MAE = metrics.mean_absolute_error(y_test, y_pred)
MSE = metrics.mean_squared_error(y_test, y_pred)

print("MAE:", MAE)
print("MSE:", MSE)
print("MAPE: ", (np.abs(y_test - y_pred)/(y_test)).mean())
```

```
MAE: 890.8168333908244
MSE: 12169888.901690237
MAPE: 0.6487123256484405
```

Dari cuplikan gambar di atas dapat dilihat bahwa MAE: 890.8168333908244, MSE: 12169888.901690237 dan MAPE : MAPE: 0.6487123256484405.

5.1.2 Evaluate Result cost prediction

Pada bagian ini dilakukan evaluate result pada *cose* prediction agar mengetahui bagaimana kinerja model yang dibangun.

```
MAE = metrics.mean_absolute_error(Y_test, Y_pred)
MSE = metrics.mean_squared_error(Y_test, Y_pred)

print('MAE:', MAE)
print('MSE:', MSE)
print("MAPE: ", (np.abs(Y_test - Y_pred)/(Y_test)).mean())
```

```
MAE: 96053.5110696544
MSE: 154033526816.72122
MAPE: 0.06898458354637245
```

Dari cuplikan gambar di atas dapat dilihat bahwa MAE: 96053.5110696544, MSE: 154033526816.72122 dan MAPE : MAPE: 0.06898458354637245.

Dari hasil evaluasi yang diperoleh dapat dibuat summary untuk Case and Cost prediction akibat penambahan rumah sakit.

Table 3. Summary Hasil Evaluasi

Menggunakan Seleksi Fitur				
	Key Performance Index	Tujuan Teknis	Hasil Model	Kondisi
Case	MAE	< 900	890.81	Terpenuhi
	MAPE	< 90%	0.6487	Terpenuhi
Cost	MAE	< 97000	96053.51	Terpenuhi
	MAPE	< 70%	0.0689	Terpenuhi

BAB 6

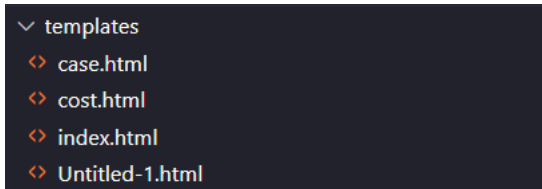
DEPLOYMENT

6.1 Plan Deployment

Dalam tahap perencanaan deployment, model yang terbentuk dari proses pemodelan akan diterapkan sesuai dengan kebutuhan data mining yang diinginkan. Langkah-langkah yang kami ambil dalam deployment yaitu :

1. Mempersiapkan interface dan Script Code

Pada tahap ini, persiapan dilakukan sebelum implementasi antarmuka (interface) dengan merancang kerangka desain HTML. Langkah ini mencakup pembuatan struktur dasar halaman web dan penyusunan elemen-elemen antarmuka. Tujuannya adalah agar proses implementasi antarmuka dengan HTML dapat dilakukan secara terstruktur. Desain antarmuka disimpan dalam folder



2. Membuat Interface HTML

- A. Case Prediction

```

plates > case.html
<body>
  <div class="wrapper">
    <div class="container">
      <div class="header">
        <div class="btn-container">
          <a href="/" class="btn btn-primary" role="button" aria-pressed="true">Kembali</a>
          <a href="/cost" class="btn btn-primary" role="button" aria-pressed="true">Cost Prediction</a>
        </div>
        <!-- Add more buttons as needed -->
      </div>

      <div class="textbox">
        <h1 class="text-3xl font-bold mb-8 text-center">Case Prediction</h1>

        <form action="/action_page.php">
          <label for="kddati2">kddati2:</label>
          <input type="text" id="kddati2" name="kddati2" required>

          <label for="tkp">tkp:</label>
          <input type="text" id="tkp" name="tkp" required>

          <label for="tkp">unit_cost:</label>
          <input type="text" id="tkp" name="tkp" required>

          <label for="peserta">peserta:</label>
          <input type="text" id="peserta" name="peserta" required>

          <label for="tipe_gabungan">tipe_gabungan:</label>
          <input type="text" id="tipe_gabungan" name="tipe_gabungan" required>

          <input type="submit" value="Submit" class="btn btn-primary">
        </form>

        <div class="col-4">
          <div class="card">
            <div class="card-header">Hasil Prediksi:</div>
            <div class="card-body">
              <p class="card-text"></p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

B. Cost Prediction

```

plates > cost.html
<body>
  <div class="wrapper">
    <div class="container">
      <div class="text-center">
        <a href="/" class="btn btn-outline-primary" role="button" aria-pressed="true"> Kembali</a>
        <a href="/case" class="btn btn-outline-success" role="button" aria-pressed="true">Case Prediction »</a>
      </div>

      <div class="textbox">
        <form method="POST" action="cost/predict">
          <div class="banner">
            <h1>Cost Prediction</h1>
          </div>
          <br>
          <label for="kddati2">kddati2:</label>
          <input type="text" id="kddati2" name="kddati2" value="">

          <label for="tkp">tkp:</label>
          <input type="text" id="tkp" name="tkp" value="">

          <label for="peserta">peserta:</label>
          <input type="text" id="peserta" name="peserta" value="">

          <label for="tipe_gabungan">tipe_gabungan:</label>
          <input type="text" id="tipe_gabungan" name="tipe_gabungan" value="">

          <input type="submit" value="Submit" style="background-color: #006622; color: #fff;">
        </form>

        <div class="col-8">
          <div class="card text-white bg-success mb-4" style="max-width: 25%;">
            <div class="card-header">Hasil Prediksi</div>
            <div class="card-body mt-3" style="padding: 10px;">
              <p class="card-text"></p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>

```

C. Index Html

```
4 }
5
6 .btn-primary {
7   background-color: #007bff;
8   color: #fff;
9 }
10
11 footer {
12   margin-top: 30px;
13   font-size: 1rem;
14   color: #6c757d;
15 }
16 </style>
17 <title>Case & Cost Prediction</title>
18 </head>
19
20 <body>
21   <div class="container">
22     <h1><strong>Case & Cost Prediction</strong></h1>
23     <p>Salah satu tugas proyek dari mata kuliah data mining</p>
24     <p>Untuk memprediksi jumlah kasus dan biaya pada sebuah daerah akibat penambahan Rumah
25       Sakit</p>
26
27     <div>
28       <a href="/case" class="btn btn-success" role="button" aria-pressed="true">Case Prediction ></a>
29       <a href="/cost" class="btn btn-primary" role="button" aria-pressed="true">Cost Prediction ></a>
30     </div>
31
32     <footer class="my-4 text-muted">0 Kelompok 5 Data Mining</footer>
33   </div>
34 </body>
35
36 </html>
37
```

3. Mempersiapkan Script dan Code

Pada proyek ini, instalasi Flask dilakukan untuk membantu menyediakan kerangka kerja web. Flask merupakan sebuah web framework yang ditulis menggunakan bahasa Python dan termasuk dalam kategori microframework. Dengan menggunakan Flask dan bahasa Python, dapat membuat situs web yang terstruktur dan dapat mengatur perilaku situs dengan lebih mudah.

```
pip install Flask
```

4. Create App.py

```

from flask import Flask, request, render_template
import pickle

import numpy

app = Flask(__name__)

model_file1 = open('modelcost.pkl', 'rb')
model1 = pickle.load(model_file1, encoding='bytes')

model_file2 = open('modelcase.pkl', 'rb')
model2 = pickle.load(model_file2, encoding='bytes')

@app.route("/", methods=['GET'])
def indexku():
    return render_template("index.html")

@app.route("/case")
def case():
    return render_template("case.html")

@app.route("/cost")
def cost():
    return render_template("cost.html")

# @app.route("case/predict", methods=["POST"])
# def cost_predict():
#     if request.method == 'POST':
#         kddati2 = float(request.form['kddati2'])
#         tkp = float(request.form['tkp'])
#         peserta = float(request.form['peserta'])
#         rs = float(request.form['rs'])
#         cost = float(request.form['cost'])
#         return render_template("case.html")
#     else:
#         return render_template("index.html")

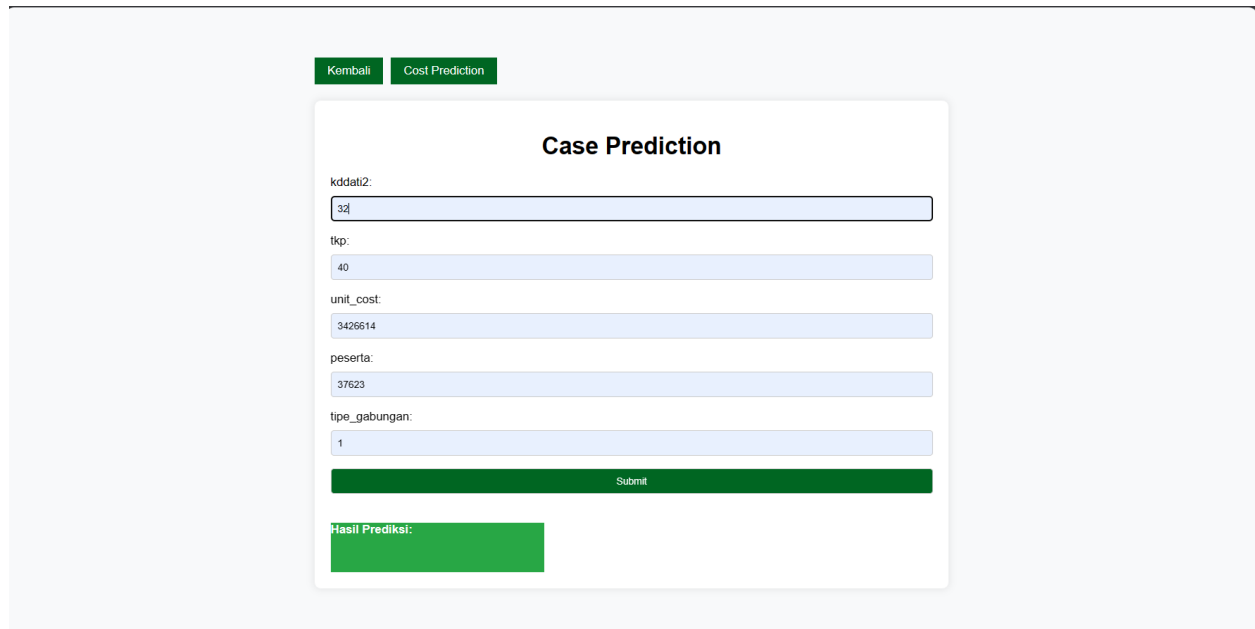
if __name__ == "__main__":

```

6.2 Web Application

Pada bagian web application dibawah, akan ditampilkan tampilan user interface dari case and cost prediction untuk data BPJS pada proyek ini

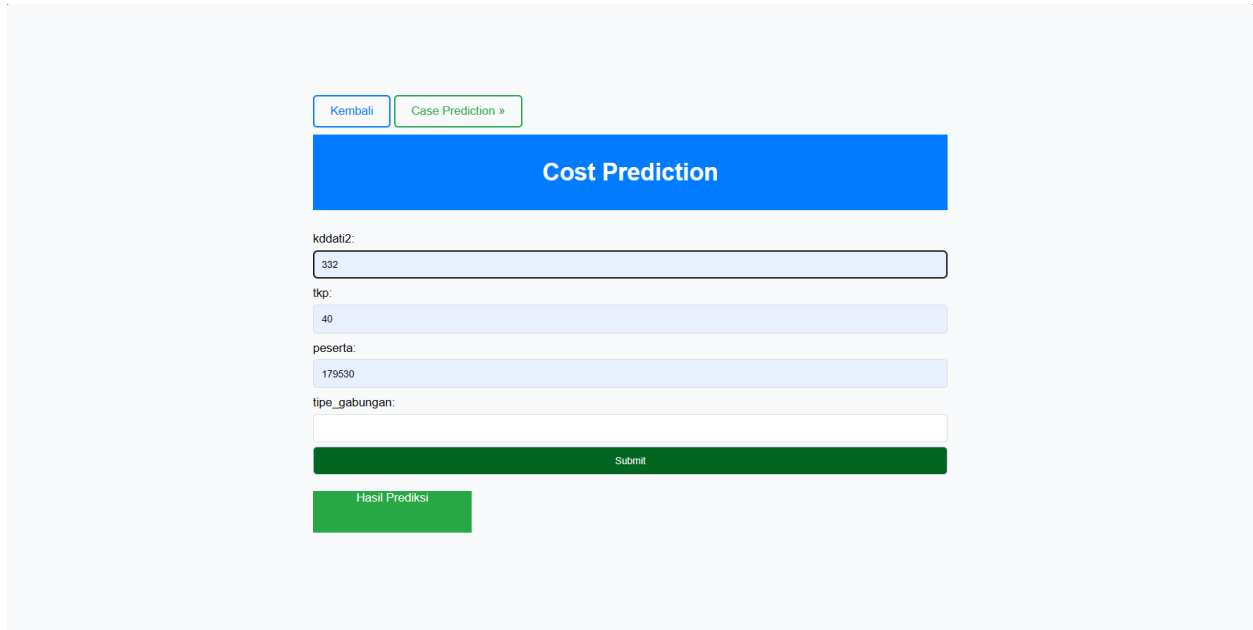
A. Case Prediction



The screenshot displays a web application interface for 'Case Prediction'. At the top, there are two green buttons: 'Kembali' (Back) and 'Cost Prediction'. The main form is titled 'Case Prediction' and contains several input fields with pre-filled values: 'kddat2:' with '32', 'tkp:' with '40', 'unit_cost:' with '3426614', 'peserta:' with '37623', and 'tipe_gabungan:' with '1'. Below these fields is a green 'Submit' button. At the bottom of the form, there is a green box labeled 'Hasil Prediksi:' (Prediction Result).

Gambar 2. Halaman Case Prediction

B. Cost Prediction



The screenshot shows a web application interface for "Cost Prediction". At the top, there are two buttons: "Kembali" (Back) and "Case Prediction »". Below these is a large blue header with the text "Cost Prediction". The form contains several input fields: "kddat2:" with the value "332", "tkp:" with the value "40", "peserta:" with the value "179530", and "tipe_gabungan:". Below the inputs is a green "Submit" button. At the bottom, there is a green box labeled "Hasil Prediksi".

Gambar 3. Cost Prediction

C. Homepage



The screenshot shows the homepage of the "Case & Cost Prediction" application. The title "Case & Cost Prediction" is displayed in large blue font. Below the title, there is a paragraph: "Salah satu tugas proyek dari mata kuliah data mining Untuk memprediksi jumlah kasus dan biaya pada sebuah daerah akibat penambahan Rumah Sakit". At the bottom, there are two buttons: "Case Prediction »" and "Cost Prediction »". A small copyright notice "© Kelompok 5 Data Mining" is visible at the very bottom.

Gambar 4. Homepage