LAPORAN

RENCANA TUGAS MAHASISWA (RTM) Ke-2

MATA KULIAH ANALISIS DATA EKSPLORATIF

"Eksplorasi Pertanyaan Dasar pada Data, Penggabungan Data (Merging), Transformasi Data, dan Deteksi Outlier"

DISUSUN OLEH:

Muhammad Aryasatya Nugroho (22083010085)

DOSEN PENGAMPU:

Tresna Maulana Fahrudin S.ST., M.T. (NIP. 199305012022031007)

PROGRAM STUDI SAINS DATA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAWA TIMUR

2023

Implementasikan teknik-teknik eksplorasi pertanyaan dasar pada data, penggabungan data, transformasi data, dan deteksi outlier untuk menyelesaikan berbagai permasalahan studi kasus pada dataset kesehatan seperti pasien kanker payudara, pasien hepatitis, dan juga dataset banking.

1. Eksplorasi Pertanyaan Dasar (Studi Kasus: Breast Cancer Dataset)

age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	Class
'40-49'	'premeno'	'15-19'	'0-2'	'yes'	'3'	'right'	'left_up'	'no'	'recurrence-events'
'50-59'	'ge40'	'15-19'	'0-2'	'no'	'1'	'right'	'central'	'no'	'no-recurrence-events'
'50-59'	'ge40'	'35-39'	'0-2'	'no'	'2'	'left'	'left_low'	'no'	'recurrence-events'
'40-49'	'premeno'	'35-39'	'0-2'	'yes'	'3'	'right'	'left_low'	'yes'	'no-recurrence-events'
'40-49'	'premeno'	'30-34'	'3-5'	'yes'	'2'	'left'	'right_up'	'no'	'recurrence-events'
…	…	…	…	…	…	…	…	…	…

a. Deskripsikan definisi masing-masing atribut pada dataset kanker payudara berdasarkan pengertian di dalam domain keilmuan medis dan referensi yang dapat dipertanggungjawabkan
   - Age: Atribut ini merujuk pada usia pasien saat diagnosis kanker payudara. Usia merupakan salah satu faktor penting dalam risiko dan pengembangan kanker payudara. Pasien yang lebih tua cenderung memiliki risiko yang lebih tinggi.
   - Menopause: Menopause mengacu pada status menopause pasien, apakah pasien telah mengalami berhenti haid atau masih dalam tahap premenopause. Status ini juga dapat mempengaruhi perkembangan dan pengobatan kanker payudara.
   - Tumor Size: Ukuran dari tumor kanker payudara dalam satuan tertentu. Ukuran faktor adalah faktor penting dalam penentuan stadium kanker dan pengambilan keputusan mengenai jenis pengobatan yang akan diberikan.
   - Inv-Nodes: Atribut ini menggambarkan sejauh mana kelenjar getah bening terlibat dalam kanker payudara.
   - Node Caps: Kapsul kelenjar getah bening mengindikasikan apakah kapsul yang melindungi telah terpengaruh oleh kanker atau tidak.
   - Degree of Malignancy: Atribut ini mencerminkan sejauh mana kanker payudara bersifat ganas atau tidak. Derajat keganasan biasanya diukur dengan skala tertentu dan dapat digunakan untuk meramalkan prognosis
   - Breast: Mengidentifikasi payudara yang terkena kanker (right atau left). Penting untuk pengobatan dan diagnosis lebih lanjut.
   - Breast Quadrant: Menunjukkan kuadran khusus dari payudara yang terkena kanker. Ini dapat membantu dalam lokalisasi tumor dan perencanaan prosedur medis.
   - Irradiation: Menunjukkan apakah pasien telah menjalani prosedur radiasi atau tidak sebagai bagian dari pengobatan kanker payudara.
   - Class: Mengklasifikasikan kasus-kasus sebagai ‘recurrence-events’ (peristiwa kambuh) atau ‘no-recurrence-events' (tanpa peristiwa kambuh). Ini adalah atribut target yang mencerminkan apakah pasien mengalami kambuh atau tidak setelah pengobatan

b. Lakukan eksplorasi pada dataset kanker payudara menggunakan Python dan jawablah pertanyaan sebagai berikut:

- Import Library dan Load Data

```
import pandas as pd
data = pd.read_csv('breast-cancer.csv',sep=';')
print(data.head())
data.dropna(inplace=True)
```

Kode diatas digunakan untuk memuat dataset kanker payudara dari file CSV, membersihkan data dari missing value, dan mencetak lima baris pertama dari dataset tersebut. Output:

```
        age  menopause tumor-size inv-nodes node-caps deg-malig   breast  \
0  '40-49'  'premeno'    '15-19'     '0-2'     'yes'       '3'  'right'
1  '50-59'     'ge40'    '15-19'     '0-2'      'no'       '1'  'right'
2  '50-59'     'ge40'    '35-39'     '0-2'      'no'       '2'   'left'
3  '40-49'  'premeno'    '35-39'     '0-2'     'yes'       '3'  'right'
4  '40-49'  'premeno'    '30-34'     '3-5'     'yes'       '2'   'left'

  breast-quad irradiat                  Class
0   'left_up'     'no'     'recurrence-events'
1   'central'     'no'  'no-recurrence-events'
2  'left_low'     'no'     'recurrence-events'
3  'left_low'    'yes'  'no-recurrence-events'
4  'right_up'     'no'     'recurrence-events'
```

- Menampilkan Informasi DataFrame

```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 277 entries, 0 to 276
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   age          277 non-null    object
 1   menopause    277 non-null    object
 2   tumor-size   277 non-null    object
 3   inv-nodes    277 non-null    object
 4   node-caps    277 non-null    object
 5   deg-malig    277 non-null    object
 6   breast       277 non-null    object
 7   breast-quad  277 non-null    object
 8   irradiat     277 non-null    object
 9   Class        277 non-null    object
dtypes: object(10)
memory usage: 21.8+ KB
None
```

Kode diatas digunakan untuk memunculkan informasi terkait dataset yang telah dimuat ke dalam DataFrame. Data diatas terdiri dari 277 baris dan 10 kolom. Setiap kolom berisi atribut tertentu yang semua kolomnya memiliki tipe data ‘object’. Output diatas menggambarkan tentang struktur dataset serta menampilkan bahwa tidak ada data yang hilang dan siap digunakan untuk tahap analisis dan eksplorasi lebih lanjut.

- Menampilkan Hasil Statistik Deskriptif

```
print(data.describe())
```

```
            age  menopause tumor-size inv-nodes node-caps deg-malig  breast  \
count       277        277        277       277       277       277     277
unique        6          3         11         7         2         3       2
top     '50-59'  'premeno'    '30-34'     '0-2'      'no'       '2'  'left'
freq         91        149         57       209       221       129     145

       breast-quad irradiat                  Class
count          277      277                    277
unique           5        2                      2
top     'left_low'     'no'  'no-recurrence-events'
freq           106      215                    196
```

Kode diatas digunakan untuk menampilkan hasil statistik deskriptif dari dataset kanker payudaara. Setiap kolom memiliki informasi penting tentang karakteristik data, termasuk jumlah data yang valid (count), jumlah nilai unik (unique), nilai yang sering muncul (top), dan frekuensi nilai yang paling sering (freq) di setiap atributnya. Dari hasil output ini, kita dapat melihat variasi dan distribusi data dalam dataset, seperti rentang usia, jenis menopausem ukuran tumor, dan lain-lain. Informasi ini juga berguna untuk pemahaman awal dataset sebelum masuk ke analisis yang lebih lanjut, kita dapat memahami pola dan hubungan dalam dataset payudara.

- Berapa banyak pasien yang berumur 50-59 tahun dengan derajat keganasan kanker payudara sebesar 2?

```
# Menggunakan fungsi crosstab
jumlah_data_per_kategori = pd.crosstab(data['age'], data['deg-malig'])
jumlah_data_per_kategori
```

deg-malig	'1'	'2'	'3'
age			
'20-29'	0	1	0
'30-39'	7	17	12
'40-49'	18	47	24
'50-59'	23	39	29
'60-69'	15	24	16
'70-79'	3	1	1

Atau dapat menggunakan:

```
KategoriPasien1 = data[(data["age"] == "'50-59'") & (data["deg-malig"] == "'2'")]
JumlahPasien1 = KategoriPasien1.shape[0]

print("Di Peroleh Jumlah Pasien Yang Berumur 50-59 Tahun Dengan Derajat Keganasan Kanker Payudara 2 Sebesar :",
      JumlahPasien1, "Pasien")
```

```
Di Peroleh Jumlah Pasien Yang Berumur 50-59 Tahun Dengan Derajat Keganasan Kanker Payudara 2 Sebesar : 39 Pasien
```

Dari kedua kode diatas dapat dilihat output kedua kode sama. Kode ini menggunakan metode pd.crosstab untuk menghitung dan menggambarkan hubungan antara usia pasien (kolom 'age') dan derajat keganasan kanker payudara (kolom 'deg-malig') dalam dataset. Hasilnya adalah tabel silang menunjukkan banyak pasien dalam setiap kelompok usia dan derajat keganasan kanker payudara. Dengan tabel ini, kita dapat dengan mudah melihat bagaimana pasien tersebar berdasarkan kedua faktor ini, yang bermanfaat dalam pemahaman awal tentang karakteristik data kanker payudara dalam dataset.
Jadi, pasien berumur 50-59 tahun dengan derajat keganasan kanker 2 sebanyak 39 pasien.

- Berapa banyak pasien dengan ukuran tumor 0-4 (mm) dengan kondisi kelenjar getah bening belum/tidak menembus kapsul dan sekitarnya (node-caps)?

```
jumlah_data_per_kategori = pd.crosstab(data['tumor-size'], data['node-caps'])
print(jumlah_data_per_kategori)
```

```
node-caps   'no'  'yes'
tumor-size
'0-4'          8      0
'10-14'       27      1
'15-19'       25      4
'20-24'       39      9
'25-29'       41     10
'30-34'       41     16
'35-39'       12      7
'40-44'       16      6
'45-49'        2      1
'5-9'          4      0
'50-54'        6      2
```

Jumlah pasien dengan ukuran tumor 0-4 mm dan kondisi node-caps 'no' adalah: 8 pasien.

- Berapa banyak pasien dengan tumor yang melakukan terapi radiasi dengan derajat keganasan kanker sebesar ‘2’ dan ‘3’?

```
jumlah_data_per_kategori = pd.crosstab(data['irradiat'], data['deg-malig'])
print(jumlah_data_per_kategori)
```

```
deg-malig  '1'  '2'  '3'
irradiat
'no'        62   98   55
'yes'        4   31   27
```

Jumlah pasien yang melakukan terapi radiasi dengan derajat keganasan kanker sebedar ‘2’ dan ‘3’ sebanyak 31 untuk keganasan kanker tingkat ‘2’ dan sebanyak 27 untuk keganasan kanker tingkat ‘3’.

- Berapa banyak pasien yang memiliki tumor yang terletak di sebelah kanan dan kiri serta tepatnya pada kuadran pusat?

```
jumlah_data_per_kategori = pd.crosstab(data['breast'], data['breast-quad'])
print(jumlah_data_per_kategori)
```

```
breast-quad  'central'  'left_low'  'left_up'  'right_low'  'right_up'
breast
'left'              11          75         34           16           9
'right'             10          31         60            7          24
```

Jumlah pasien yang memiliki tumor di sebelah kanan tepatnya pada kuadran pusat sebanyak 10 pasien, sementara pasien yang memiliki tumor terletak di sebelah kiri tepatnya pada kuadran pusat berjumlah 11 pasien.

- Berapa banyak pasien yang sedang premenopause dengan kelenjar getah bening yang mengandung kanker payudara pada range 6-8?

```
jumlah_data_per_kategori = pd.crosstab(data['menopause'], data['inv-nodes'])
print(jumlah_data_per_kategori)
```

```
inv-nodes  '0-2'  '12-14'  '15-17'  '24-26'  '3-5'  '6-8'  '9-11'
menopause
'ge40'        93        1        3        1     14     10       1
'lt40'         5        0        0        0      0      0       0
'premeno'    111        2        3        0     20      7       6
```

Jumlah pasien yang sedang premenopause dengan kelenjar getah bening yang mengandung kanker payudara pada range 6-8 berjumlah 7 pasien.

- Berapa banyak pasien yang telah melakukan terapi radiasi, tetapi masih ada kemungkinan terjadi kekambuhan ulang?

```
jumlah_data_per_kategori = pd.crosstab(data['irradiat'], data['Class'])
print(jumlah_data_per_kategori)
```

```
Class     'no-recurrence-events'  'recurrence-events'
irradiat
'no'                         164                   51
'yes'                         32                   30
```

Jumlah pasien yang telah melakukan terapi radiasi tapi masih terdaat kemungkinan kembuh ulang sejumlah 30 pasien.

- Berapa banyak pasien yang masih berumur 30-39 tahun dengan kondisi kelenjar getah bening telah menembus kapsul dan sekitarnya (node-caps)?

```
jumlah_data_per_kategori = pd.crosstab(data['node-caps'], data['age'])
print(jumlah_data_per_kategori)
```

```
age        '20-29'  '30-39'  '40-49'  '50-59'  '60-69'  '70-79'
node-caps
'no'             1       28       71       70       46        5
'yes'            0        8       18       21        9        0
```

Jumlah pasien yang berumur 30-39 tahun dengan kondisi kelenjar getah bening menembus kapsul dan sekitarnya (node-caps) berjumlah 8 pasien.

- Berapa banyak pasien yang menopause di atas umur 40 tahun, tetapi tidak melakukan terapi radiasi?

```
jumlah_data_per_kategori = pd.crosstab(data['menopause'], data['irradiat'])
print(jumlah_data_per_kategori)
```

```
irradiat   'no'  'yes'
menopause
'ge40'       99     24
'lt40'        5      0
'premeno'   111     38
```

Jumlah pasien yang menopause diatas umur 40 tahun, tetapi tidak melakukan terapi radiasi berjumlah 99 pasien.

- Berapa banyak pasien dengan ukuran tumor sebesar 50-54 (mm) dengan kelenjar getah bening aksila yang mengandung kanker payudara metastatik sebesar 0-2 (mm)?

```
jumlah_data_per_kategori = pd.crosstab(data['inv-nodes'], data['tumor-size'])
print(jumlah_data_per_kategori)
```

```
tumor-size  '0-4'  '10-14'  '15-19'  '20-24'  '25-29'  '30-34'  '35-39'  \
inv-nodes
'0-2'           8       26       26       34       39       36       13
'12-14'         0        0        1        0        1        1        0
'15-17'         0        0        1        0        1        1        2
'24-26'         0        0        0        1        0        0        0
'3-5'           0        1        0       12        6       10        0
'6-8'           0        1        1        1        3        7        1
'9-11'          0        0        0        0        1        2        3

tumor-size  '40-44'  '45-49'  '5-9'  '50-54'
inv-nodes
'0-2'            14        2      4        7
'12-14'           0        0      0        0
'15-17'           1        0      0        0
'24-26'           0        0      0        0
'3-5'             5        0      0        0
'6-8'             2        1      0        0
'9-11'            0        0      0        1
```

Jumlah pasien dengan ukuran tumor 50-54 (mm) dengan kelenjar getah bening aksila yang mengandung kanker payudara metastatic sebesar 0-2 (mm) berjumlah 7 pasien.

- Berapa banyak pasien dengan kelenjar getah bening aksila yang mengandung kanker payudara metastatic sebesar 15-17 (mm) dan dengan kondisi kelenjar getah bening belum/tidak menembus kapsul dan sekitarnya (node-caps)?

```
jumlah_data_per_kategori = pd.crosstab(data['inv-nodes'], data['node-caps'])
print(jumlah_data_per_kategori)
```

```
node-caps  'no'  'yes'
inv-nodes
'0-2'       200      9
'12-14'       1      2
'15-17'       1      5
'24-26'       0      1
'3-5'        15     19
'6-8'         3     14
'9-11'        1      6
```

Jumlah pasien dengan kelenjar getah bening aksila yang mengandung kanker payudara metastatic sebesar 15-17 (mm) dan dengan kondisi kelenjar getah bening belum/tidak menembus kapsul dan sekitarnya (node-caps) berjumlah 1 pasien.

2. Penggabungan Data Menggunakan Inner Join, Left Join, Right Join, dan Outer Join (Studi Kasus: Breast Cancer Dataset)

PatientID	Tumor Size
1	'15-19'
3	'35-39'
5	'30-34'
7	'40-44'
9	'0-4'
11	'25-29'
15	'25-29'
17	'10-14'
19	'40-44'
21	'40-44'
...	...

PatientID	Tumor Size
4	'35-39'
6	'25-29'
8	'10-14'
12	'15-19'
14	'25-29'
16	'20-24'
18	'15-19'
22	'15-19'
24	'15-19'
26	'10-14'
...	...

PatientID	Breast Quandrant
1	'left_up'
3	'left_low'
7	'left_up'
9	'right_low'
11	'left_low'
13	'central'
17	'left_up'
19	'left_up'
21	'left_up'
23	'left_up'
...	...

PatientID	Breast Quadrant
2	'central'
6	'left_up'
8	'left_up'
10	'left_up'
12	'left_up'
14	'left_up'
16	'central'
18	'left_up'
22	'left_low'
24	'left_low'
...	....

a. Implementasikan penggabungan data (merge) menggunakan Inner Join, Left Join, Right Join, dan Outer Join menggunakan Python untuk menjawab pertanyaan berikut:

- Penggabungan Dataset Menjadi DataFrame:

```python
import pandas as pd

tumor_size_odd = pd.read_csv('df1tumor_size_odd.csv', sep=';')
breast_quad_odd = pd.read_csv('df1breastquad_odd.csv', sep=';')
breast_quad_even = pd.read_csv('df2breastquad_even.csv', sep=';')
tumor_size_even = pd.read_csv('df2tumor_size_even.csv', sep=';')


print("Dataset tumor_size_odd:")
print(tumor_size_odd)

print("\nDataset breast_quad_odd:")
print(breast_quad_odd)

print("\nDataset breast_quad_even:")
print(breast_quad_even)

print("\nDataset tumor_size_even:")
print(tumor_size_even)
```

Kode diatas digunakan untuk memuat dan menampilkan empat dataset berbentuk CSV. Setelah itu ditampilkan output yang mencetak masing-masing dari isi dataset diatas untuk dapat dipantau dan dianalisis lebih lanjut seperi berikut:

```
Dataset tumor_size_odd:
     PatientID tumor-size
0            1    '15-19'
1            3    '35-39'
2            5    '30-34'
3            7    '40-44'
4            9      '0-4'
..         ...        ...
115        265    '20-24'
116        269    '35-39'
117        271    '20-24'
118        275    '30-34'
119        277    '40-44'

[120 rows x 2 columns]

Dataset breast_quad_odd:
     PatientID  breast-quad
0            1    'left_up'
1            3   'left_low'
2            7    'left_up'
3            9  'right_low'
4           11   'left_low'
..         ...          ...
114        269   'right_up'
115        271   'right_up'
116        273   'left_low'
117        275   'right_up'
118        277   'right_up'

[119 rows x 2 columns]

Dataset breast_quad_even:
     PatientID breast-quad
0            2   'central'
1            6   'left_up'
2            8   'left_up'
3           10   'left_up'
4           12   'left_up'
..         ...         ...
113        264   'left_up'
114        266  'left_low'
115        268   'left_up'
116        270  'left_low'
117        274  'left_low'

[118 rows x 2 columns]

Dataset tumor_size_even:
     PatientID tumor-size
0            4    '35-39'
1            6    '25-29'
2            8    '10-14'
3           12    '15-19'
4           14    '25-29'
..         ...        ...
113        268    '25-29'
114        270    '30-34'
115        272    '15-19'
116        274    '25-29'
117        276    '15-19'

[118 rows x 2 columns]
```

- Gabungkan Kedua Dataset Tumor-Size:

```
1 data_tumor_size_odd = pd.read_csv('df1tumor_size_odd.csv', sep=';')
2 data_breast_quad_odd = pd.read_csv('df1breastquad_odd.csv', sep=';')
3 data_breast_quad_even = pd.read_csv('df2breastquad_even.csv', sep=';')
4 data_tumor_size_even = pd.read_csv('df2tumor_size_even.csv', sep=';')
5
6 gabung_dfsize = pd.concat([data_tumor_size_odd,data_tumor_size_even]).sort_values('PatientID')
7 gabung_dfsize
```

	PatientID	tumor-size
0	1	'15-19'
1	3	'35-39'
0	4	'35-39'
2	5	'30-34'
1	6	'25-29'
...	...	...
115	272	'15-19'
116	274	'25-29'
118	275	'30-34'
117	276	'15-19'
119	277	'40-44'

238 rows × 2 columns

Kode ini digunakan untuk menggabungkan dua dataset ukuran tumor ('data_tumor_size_odd' dan 'data_tumor_size_even') menjadi satu dataset tunggal ('gabung_dfsize') berdasarkan kolom 'PatientID'.

- Penggabungan Dua Dataset Breast-Quad

```
gabung_dfquad = pd.concat([data_breast_quad_odd, data_breast_quad_even]).sort_values("PatientID")
gabung_dfquad
```

	PatientID	breast-quad
0	1	'left_up'
0	2	'central'
1	3	'left_low'
1	6	'left_up'
2	7	'left_up'
...	...	...
115	271	'right_up'
116	273	'left_low'
117	274	'left_low'
117	275	'right_up'
118	277	'right_up'

237 rows × 2 columns

Kode diatas bertujuan serupa seperti kode sebelumnya, namun digunakan untuk merging dua dataset terkait kuadran payudara ('data_breast_quad_odd' dan 'data_breast_quad_even') menjadi satu dataset tunggal yang diberi nama 'gabung_dfquad'. Penggabungan kedua dataset tersebut dimaksudkan agar memudahkan analisis eksplorasi lebih lanjut.

- Berapa banyak record pasien yang memiliki nilai atribut ukuran tumor dan nilai kuadran kanker payudara secara lengkap keduanya?

```
total_lengkap = inner_join.shape[0]
print(f"Jumlah record pasien dengan kedua atribut lengkap : {total_lengkap}")
```

```
Jumlah record pasien dengan kedua atribut lengkap : 203
```

Kode diatas digunakan untuk Menghitung jumlah pasien dalam dataset yang memiliki kedua atribut (kolom) lengkap. Hasil akan disimpan dalam variable total_lengkap, variable tersebut merupakan jumlah pasien yang data-data lengkapnya telah dihitung dengan mengambil jumlah baris dari DataFrame hasil inner join antara dua dataset. Dalam hal ini, mencari pasien yang memiliki data lengkap guna untuk analisis atau pemodelan data lebih lanjut.

- Berapa banyak record pasien yang memiliki nilai atribut ukuran tumor, tetapi tidak memiliki nilai kuadran kanker payudara? serta berapa banyak record nilai atribut kuadran kanker payudara yang NaN tersebut?

```
size_only = left_join.shape[0]
print("Jumlah tumor size tanpa breast-quad =", size_only)
jumlah_kuadran_nan = left_join["breast-quad"].isnull().sum()
print("Jumlah kuadran yang NaN =", jumlah_kuadran_nan)
```

```
Jumlah tumor size tanpa breast-quad = 238
Jumlah kuadran yang NaN = 35
```

Kode diatas menghitung jumlah pasien yang hanya memiliki atribut "tumor size" tanpa atribut "breast-quad" dalam dataset dan menyimpannya dalam variabel size_only. Kemudian, kode menghitung jumlah kuadran payudara yang memiliki nilai yang hilang (NaN) dalam dataset dan menyimpannya dalam variabel jumlah_kuadran_nan. Ini membantu dalam mengidentifikasi berapa banyak data yang tidak lengkap dalam dataset.

- Berapa banyak record pasien yang tidak memiliki nilai atribut ukuran tumor, tetapi memiliki nilai kuadran kanker payudara? serta berapa banyak record nilai atribut ukuran tumor yang NaN tersebut?

```
breast_quad_only = right_join.shape[0]
print(f"Yang tidak memiliki atribut tumor size: {breast_quad_only}")
ukuran_nan = right_join["tumor-size"].isnull().sum()
print(f"Jumlah NaN atribut size tersebut: {ukuran_nan}")
```

```
Yang tidak memiliki atribut tumor size: 237
Jumlah NaN atribut size tersebut: 34
```

Kode tersebut menghitung jumlah pasien yang hanya memiliki atribut "kuadran payudara" tanpa atribut "ukuran tumor" dalam dataset dan menyimpannya dalam variabel breast_quad_only. Selanjutnya, kode menghitung jumlah data ukuran tumor yang memiliki nilai null (NaN) dalam dataset dan menyimpannya dalam variabel ukuran_nan. Ini membantu dalam mengidentifikasi pasien yang hanya memiliki data kuadran payudara tanpa ukuran tumor, serta seberapa banyak data ukuran tumor yang tidak lengkap dalam dataset.

- Jika menggunakan Outer Join, Berapa banyak record pasien yang NaN pada masing-masing atribut, baik pada atribut ukuran tumor maupun kuadran kanker payudara?

```
pasien_nan = outer_join.isnull().sum()
print ((f"Jumlah record pasien yang NaN pada masing-masing atribut:\n {pasien_nan}"))
```

```
Jumlah record pasien yang NaN pada masing-masing atribut:
 PatientID       0
tumor-size      34
breast-quad     35
dtype: int64
```

Kode ini menghitung jumlah data yang memiliki nilai null (NaN) dalam setiap atribut (kolom) dalam dataset setelah mengalami outer join (gabungan data yang mempertahankan semua baris dari kedua dataset). Hasilnya disimpan dalam variabel pasien_nan. Ini dapat membantu memahami sejauh mana nilai yang hilang pada setiap atributnya, yang penting dalam analisis data dan pemrosesan lanjutan.

3. Transformasi Data: Duplikasi Data, Missing Values, dan Imputasi Missing Values

#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	Age	Sex	Steroid	Antivirals	Fatique	Malaise	Anorexia	Liver Big	Liver Firm	Spleen Palpable	Speiders	Ascites	Varices	Bilirubin	Alk Phosphat	SGOT	Albumin	Protime	Histology	CLASS
1	30	2	1	2	2	2	2	1	2	2	2	2	2	1	85	18	4	?	1	2
2	50	1	1	2	1	2	2	1	2	2	2	2	2	0,9	135	42	3,5	?	1	2
3	78	1	2	2	1	2	2	2	2	2	2	2	2	0,7	96	32	4	?	1	2
4	31	1	?	1	2	2	2	2	2	2	2	2	2	0,7	46	52	4	80	1	2
5	34	1	2	2	2	2	2	2	2	2	2	2	2	1	?	200	4	?	1	2

a. Deskripsikan definisi masing-masing atribut pada dataset hepatitis berdasarkan pengertian di dalam domain keilmuan medis
   - Age: Mengindikasikan usia pasien yang terkena hepatitis.
   - Sex: Mengidentifikasi jenis kelamin pasien, yaitu apakah pasien adalah pria (1) atau wanita (2). Jenis kelamin dapat memengaruhi cara hepatitis mempengaruhi individu dan respon terhadap perawatan.
   - Steroid: Penggunaan steroid oleh pasien. Steroid bisa digunakan dalam pengobatan hepatitis tertentu dan dapat mempengaruhi perkembangan penyakit.
   - Antivirals: Mengindikasikan apakah pasien menerima pengobatan antiviral (1) atau tidak (2). Pengobatan antiviral sering digunakan dalam penanganan hepatitis virus.
   - Fatigue: Menggambarkan tingkat kelelahan atau kelelahan yang dialami oleh pasien. Kelelahan adalah salah satu gejala umum hepatitis.
   - Malaise: Mengukur tingkat malaise atau ketidaknyamanan umum yang dirasakan oleh pasien. Ini juga merupakan gejala umum hepatitis.
   - Anorexia: Mengindikasikan apakah pasien mengalami hilangnya nafsu makan (anoreksia). Hilangnya nafsu makan adalah gejala lain yang sering terkait dengan hepatitis.
   - Liver Big: Apakah pasien memiliki hati yang membesar (hepatomegali). Hepatomegali dapat menjadi tanda penyakit hati, termasuk hepatitis.
   - Liver Firm: Mengindikasikan apakah hati pasien terasa keras saat dipegang. Ini dapat menunjukkan kerusakan hati yang terkait dengan hepatitis.
   - Spleen Palpable: Menunjukkan apakah limpa pasien dapat dirasakan (palpable) atau tidak. Perbesaran limpa (splenomegali) dapat terjadi pada beberapa kasus hepatitis.
   - Speiders: Mengukur tanda-tanda spider angiomas pada pasien, yang adalah pembuluh darah yang membesar di kulit. Ini bisa menjadi tanda kerusakan hati.
   - Ascites: Mengindikasikan apakah pasien memiliki akumulasi cairan di perut (ascites). Ascites adalah komplikasi serius yang dapat terjadi pada hepatitis.
   - Varices: Apakah pasien memiliki varises esofagus (pelebaran pembuluh darah di kerongkongan). Varises esofagus dapat terjadi sebagai akibat dari kerusakan hati yang parah.
   - Bilirubin: Mengukur kadar bilirubin dalam darah pasien. Bilirubin adalah produk pemecahan hemoglobin yang dapat meningkat dalam kasus kerusakan hati.
   - Alk Phosphate: Mengukur kadar fosfatase alkali dalam darah pasien. Peningkatan fosfatase alkali bisa menjadi tanda kerusakan hati.
   - SGOT (Serum Glutamic Oxaloacetic Transaminase): Mengukur kadar SGOT dalam darah pasien. Peningkatan SGOT bisa menunjukkan kerusakan hati.
   - Albumin: Mengukur kadar albumin dalam darah pasien. Albumin adalah protein yang diproduksi oleh hati dan dapat menurun dalam kasus penyakit hati.
   - Protime (Prothrombin Time): Mengukur waktu pembekuan darah pasien. Perpanjangan waktu pembekuan darah dapat terjadi pada penyakit hati yang parah.
   - Histology: Hasil pemeriksaan jaringan hati pasien melalui histologi. Ini adalah metode untuk mendeteksi perubahan patologis dalam hati.
   - CLASS: Ini adalah atribut target yang mengidentifikasi kelas pasien (1 atau 2) sebagai kasus hepatitis (1) atau non-hepatitis (2).

b. Lakukan transformasi data menggunakan Python dengan Langkah-langkah seperti berikut:

- Import Library dan Load Data

```
import pandas as pd
import numpy as np

hepatitis = pd.read_excel('Hepatitis.xlsx', header=1)
```

```
hepatitis = hepatitis.drop(columns=['Unnamed: 0'])
hepatitis
```

	Age	Sex	Steroid	Antivirals	Fatique	Malaise	Anorexia	Liver Big	Liver Firm	Spleen Palpable	Speiders	Ascites	Varices	Bilirubin	Alk Phosphate	SGOT	Albumi
0	30	2	1	2	2	2	2	1	2	2	2	2	2	1	85	18	
1	50	1	1	2	1	2	2	1	2	2	2	2	2	0.9	135	42	3.
2	78	1	2	2	1	2	2	2	2	2	2	2	2	0.7	96	32	
3	31	1	?	1	2	2	2	2	2	2	2	2	2	0.7	46	52	
4	34	1	2	2	2	2	2	2	2	2	2	2	2	1	?	200	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
150	46	1	2	2	1	1	1	2	2	2	1	1	1	7.6	?	242	3.
151	44	1	2	2	1	2	2	2	1	2	2	2	2	0.9	126	142	4.
152	61	1	1	2	1	1	2	1	1	2	1	2	2	0.8	75	20	4.
153	53	2	1	2	1	2	2	2	2	1	1	2	1	1.5	81	19	4.
154	43	1	2	2	1	2	2	2	2	1	1	1	2	1.2	100	19	3.

Kode diatas digunakan untuk membaca sebuah file Excel yang kemudian disimpan dalan variable ‘hepatitis’. Selanjutnya, kode menghilangkan kolom 'Unnamed: 0' dari dataset dengan menggunakan perintah hepatitis = hepatitis.drop(columns=['Unnamed: 0']). Ini dilakukan untuk menghapus kolom yang mungkin tidak relevan atau tidak memiliki nama yang jelas dalam dataset.

- Lakukan pemeriksaan apakah terdapat duplikasi data (row) pada dataset hepatitis

```
cek_duplikat = hepatitis[hepatitis.duplicated()]

if not cek_duplikat.empty:
    print("Terdapat duplikasi data:")
    print(cek_duplikat)
else:
    print("Tidak ada duplikasi data")
```

```
Tidak ada duplikasi data
```

Diperoleh hasil, tidak terdapat duplikasi data dalam dataset diatas.

- Lakukan analisis deskriptif berapa banyak atribut yang memiliki missing values

```
missing_values = hepatitis.isnull().sum()
print("Jumlah missing values untuk setiap atribut:")
missing_values
```

Kode di atas digunakan untuk menghitung jumlah nilai yang hilang (missing values) dalam dataset hepatitis untuk setiap atribut (kolom). Proses ini dilakukan dengan menggunakan metode isnull().sum(). Ditampilkan hasil sebagai berikut:

```
Jumlah missing values untuk setiap atribut:

Age                0
Sex                0
Steroid            1
Antivirals         0
Fatique            1
Malaise            1
Anorexia           1
Liver Big         10
Liver Firm        11
Spleen Palpable    5
Speiders           5
Ascites            5
Varices            5
Bilirubin          6
Alk Phosphate     29
SGOT               4
Albumin           16
Protime           67
Histology          0
CLASS              0
dtype: int64
```

- Lakukan imputasi missing values dengan berbagai teknik yang efektif dan baik untuk mengisi atribut-atribut yang NaN, misalnya menggunakan mean, median, modus, clustering, regression, maupun metode taksiran dan prediksi lainnya
  a) Menggunakan Metode Mean, Median, Modus

```
mean = ["Alk Phosphate", "SGOT", "Albumin", "Protime", "Bilirubin"]
modus = ["Steroid", "Fatique", "Anorexia", "Liver Big", "Liver Firm", "Spleen Palpable", "Speiders", "Malaise", "Anorexia
```

```
for i in mean:
    hepatitis[i].fillna(hepatitis[i].mean(), inplace=True)
for j in modus:
    modus_col = hepatitis[j].mode()[0]
    hepatitis[j].fillna(modus_col, inplace=True)

for k in hepatitis.columns:
    hepatitis[k] = hepatitis[k].round(1)
```

Kode di atas digunakan untuk mengisi nilai-nilai yang hilang dalam dataset hepatitis dengan metode yang sesuai berdasarkan jenis atributnya. Atribut-atribut yang berisi data numerik seperti "Alk Phosphate", "SGOT", "Albumin", "Protime", dan "Bilirubin" diisi dengan nilai rata-rata dari masing-masing atribut, sementara atribut-atribut kategorikal seperti "Steroid", "Fatique", "Anorexia", dan lainnya diisi dengan nilai modus (nilai yang paling sering muncul) dari masing-masing atribut. Selain itu, semua nilai dalam dataset dibulatkan menjadi satu angka desimal. Dengan demikian, kode ini memproses dan mengolah data untuk memastikan bahwa dataset menjadi lebih lengkap dan siap digunakan dalam analisis data lebih lanjut.

b) Menggunakan Metode Regresi

```
from sklearn.linear_model import LinearRegression

data_lengkap = hepatitis.dropna(subset = ["Protime", "Bilirubin", "Alk Phosphate", "SGOT"])
data_imputasi = hepatitis[hepatitis["Protime"].isnull()]

model_regresi = LinearRegression()
model_regresi.fit(data_lengkap[["Bilirubin", "Alk Phosphate", "SGOT"]], data_lengkap["Protime"])

nilai_prediksi = model_regresi.predict(data_imputasi[["Bilirubin", "Alk Phosphate", "SGOT"]])

data_imputasi["Protime"] = nilai_prediksi
hepatitis.loc[data_imputasi.index] = data_imputasi
```

Kode diatas menggunakan metode regresi linier untuk mengisi nilai-nilai yang hilang dalam atribut protime pada dataset. Langkah pertama, data lengkap dan data yang memerlukan imputasi dipisahkan. Kemudian, sebuah model regresi linier dibuat dengan atribut seperti kode diatas untuk memprediksi nilai 'Protime'. Model ini digunakan untuk mengisi nilai yang hilang pada data imputasi, dan hasilnya disimpan Kembali ke dalam dataset. Tujuannya untuk mengisi nilai yang hilang dengan perkiraan yang masuk akal berdasarkan hubungan linier antar atribut, sehingga dataset tetap bisa digunakan untuk analisis lebih lanjut.

- Menampilkan DataFrame Setelah Mengimputasi Missing Values

Age	Sex	Steroid	Antivirals	Fatique	Malaise	Anorexia	Liver Big	Liver Firm	Spleen Palpable	Speiders	Ascites	Varices	Bilirubin	Alk Phosphate	SGOT	Albumin	Protime	Histology
30	2	1.0	2	2.0	2.0	2.0	1.0	2.0	2.0	2.0	2.0	2.0	1.0	85.0	18.0	4.0	61.9	
50	1	1.0	2	1.0	2.0	2.0	1.0	2.0	2.0	2.0	2.0	2.0	0.9	135.0	42.0	3.5	61.9	
78	1	2.0	2	1.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.7	96.0	32.0	4.0	61.9	
31	1	2.0	1	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.7	46.0	52.0	4.0	80.0	
34	1	2.0	2	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	1.0	105.3	200.0	4.0	61.9	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
46	1	2.0	2	1.0	1.0	1.0	2.0	2.0	2.0	1.0	1.0	1.0	7.6	105.3	242.0	3.3	50.0	
44	1	2.0	2	1.0	2.0	2.0	2.0	1.0	2.0	2.0	2.0	2.0	0.9	126.0	142.0	4.3	61.9	
61	1	1.0	2	1.0	1.0	2.0	1.0	1.0	2.0	1.0	2.0	2.0	0.8	75.0	20.0	4.1	61.9	
53	2	1.0	2	1.0	2.0	2.0	2.0	2.0	1.0	1.0	2.0	1.0	1.5	81.0	19.0	4.1	48.0	
43	1	2.0	2	1.0	2.0	2.0	2.0	2.0	1.0	1.0	1.0	2.0	1.2	100.0	19.0	3.1	42.0	

- Kembali lakukan pengecekan missing values

```
1 missing_values = hepatitis.isnull().sum()
2 print("Jumlah missing values untuk setiap atribut:")
3 print(missing_values)
```

```
Jumlah missing values untuk setiap atribut:
Age                0
Sex                0
Steroid            0
Antivirals         0
Fatique            0
Malaise            0
Anorexia           0
Liver Big          0
Liver Firm         0
Spleen Palpable    0
Speiders           0
Ascites            0
Varices            0
Bilirubin          0
Alk Phosphate      0
SGOT               0
Albumin            0
Protime            0
Histology          0
CLASS              0
dtype: int64
```

Output yang diperoleh, semua atribut dalam dataset tidak memiliki missing values (0 missing values). Ini menunjukkan dataset tidak memiliki nilai yang hilang dalam atribut-atributnya, yang artinya data diatas sudah ideal untuk dilakukan analisis data lebih lanjut.

4. Deteksi Outlier: InterQuartile Range dan BoxPlot

ATM Name	Transaction Date	No Of Withdrawals	Total amount Withdrawn (in Rupee)	Total amount Withdrawn (in Rupiah)	Weekday
Big Street ATM	01/01/11	50	123800	23136982	Saturday
Mount Road ATM	01/01/11	253	767900	143512831	Saturday
Airport ATM	01/01/11	98	503400	94080426	Saturday
KK Nagar ATM	01/01/11	265	945300	176667117	Saturday
Christ College ATM	01/01/11	74	287700	53768253	Saturday
Big Street ATM	02/01/11	17	52800	9867792	Sunday

…..

a. Deskripsikan definisi masing-masing atribut pada dataset ATM Transcation berdasarkan pengertian di dalam domain keilmuan banking dan referensi yang dapat dipertanggungjawabkan

- ATM Name: Nama ATM tempat transaksi dilakukan. Dalam domain perbankan, ATM (Automated Teller Machine) adalah mesin yang memungkinkan nasabah untuk melakukan transaksi perbankan seperti penarikan tunai, cek saldo, dan transfer dana.
- Transaction Date: Mencakup tanggal transaksi. Dalam konteks perbankan, tanggal transaksi adalah tanggal ketika nasabah melakukan operasi perbankan tertentu di ATM.
- No Of Withdrawals: Jumlah transaksi penarikan tunai yang dilakukan. Atribut ini mencatat berapa kali nasabah menarik uang tunai dari ATM pada tanggal tertentu.
- Total amount Withdrawn (in Rupee): Total jumlah uang yang ditarik dalam mata uang Rupee (mata uang resmi India). Atribut ini mencerminkan total nilai uang tunai yang ditarik oleh nasabah pada tanggal tersebut.
- Total amount Withdrawn (in Rupiah): Ini adalah total jumlah uang yang ditarik dalam mata uang Rupiah. Meskipun dataset tidak menyebutkan negara, atribut ini mungkin mencerminkan konversi jumlah uang tunai yang ditarik dalam mata uang Rupiah jika transaksi dilakukan di negara yang menggunakan Rupiah sebagai mata uang.
- Weekday: Ini adalah hari dalam seminggu ketika transaksi dilakukan. Atribut ini mencatat hari kapan nasabah melakukan transaksi di ATM.

b. Lakukan analisis deskiriptif dan deteksi outlier menggunakan Python dan menjawab pertanyaan berikut:

- Import Library dan Load Dataset

```
import pandas as pd
import matplotlib.pyplot as plt

atm = pd.read_csv('ATM-Transaction.csv', sep=';')
atm
```

Kode diatas digunakan untuk menampilkan dataset ke dalam bentuk DataFrame. Output:

	ATM Name	Transaction Date	No Of Withdrawals	Total amount Withdrawn (Rupee)	Total amount Withdrawn (Rupiah)	Weekday
0	Big Street ATM	01/01/11	50	123800	23136982	Saturday
1	Mount Road ATM	01/01/11	253	767900	143512831	Saturday
2	Airport ATM	01/01/11	98	503400	94080426	Saturday
3	KK Nagar ATM	01/01/11	265	945300	176667117	Saturday
4	Christ College ATM	01/01/11	74	287700	53768253	Saturday
...	...	...	...	...	...	...
11584	Big Street ATM	29/09/17	137	468800	87614032	FRIDAY
11585	Mount Road ATM	29/09/17	79	305100	57020139	FRIDAY
11586	Airport ATM	29/09/17	117	709900	132673211	FRIDAY
11587	KK Nagar ATM	29/09/17	76	408700	76381943	FRIDAY
11588	Christ College ATM	29/09/17	143	700400	130897756	FRIDAY

11589 rows × 6 columns

- Bank mana yang teramai dan tersepi bagi nasabah untuk melakukan penarikan uang?

```
total_withdrawals = atm.groupby('ATM Name')['No Of Withdrawals'].sum()

# Temukan ATM dengan jumlah penarikan tertinggi
atm_teramai = total_withdrawals.idxmax()
atm_tersepi = total_withdrawals.idxmin()
jumlah_penarikan_terbanyak = total_withdrawals.max()
jumlah_penarikan_terendah = total_withdrawals.min()


print("Jumlah penarikan tiap bank:\n")
for atm_name, total_withdrawal in total_withdrawals.items():
    print(f"{atm_name}: {total_withdrawal} penarikan")

print(f"Penarikan ATM teramai: {atm_teramai} dengan jumlah {jumlah_penarikan_terbanyak} penarikan")
print(f"Penarikan ATM tersepi: {atm_tersepi} dengan jumlah {jumlah_penarikan_terendah} penarikan")

```

```
Jumlah penarikan tiap bank:

Airport ATM: 204709 penarikan
Big Street ATM: 207062 penarikan
Christ College ATM: 291207 penarikan
KK Nagar ATM: 401858 penarikan
Mount Road ATM: 324564 penarikan
Penarikan ATM teramai: KK Nagar ATM dengan jumlah 401858 penarikan
Penarikan ATM tersepi: Airport ATM dengan jumlah 204709 penarikan
```

Kode tersebut menghitung jumlah penarikan tunai (No Of Withdrawals) untuk setiap ATM (ATM Name) dalam dataset. Selanjutnya, kode menemukan ATM dengan jumlah penarikan tertinggi (teramai) dan ATM dengan jumlah penarikan terendah (tersepi), serta mencetak daftar jumlah penarikan tunai tiap ATM beserta ATM yang paling ramai dan sepi beserta jumlah penarikannya. Ini berguna untuk menganalisis aktivitas penarikan tunai di berbagai ATM dan

mengidentifikasi ATM yang paling banyak digunakan serta yang paling sedikit digunakan dalam dataset.

- Pada hari apa ATM paling sering dan pada hari apa yang paling jarang dikunjungi oleh nasabah untuk melakukan penarikan uang?

```
atm.replace('SUNDAY', 'Sunday', inplace=True)
atm.replace('MONDAY', 'Monday', inplace=True)
atm.replace('TUESDAY', 'Tuesday', inplace=True)
atm.replace('WEDNESDAY', 'Wednesday', inplace=True)
atm.replace('THURSDAY', 'Thursday', inplace=True)
atm.replace('FRIDAY', 'Friday', inplace=True)
atm.replace('SATURDAY', 'Saturday', inplace=True)
atm
```

Kode diatas digunakan untuk mengubah kata yang mewakili hari dalam seminggu agar menjadi satu versi untuk pemrosesan selanjutnya. Output

	ATM Name	Transaction Date	No Of Withdrawals	Total amount Withdrawn (Rupee)	Total amount Withdrawn (Rupiah)	Weekday
0	Big Street ATM	01/01/11	50	123800	23136982	Saturday
1	Mount Road ATM	01/01/11	253	767900	143512831	Saturday
2	Airport ATM	01/01/11	98	503400	94080426	Saturday
3	KK Nagar ATM	01/01/11	265	945300	176667117	Saturday
4	Christ College ATM	01/01/11	74	287700	53768253	Saturday
...	...	...	...	...	...	...
11584	Big Street ATM	29/09/17	137	468800	87614032	Friday
11585	Mount Road ATM	29/09/17	79	305100	57020139	Friday
11586	Airport ATM	29/09/17	117	709900	132673211	Friday
11587	KK Nagar ATM	29/09/17	76	408700	76381943	Friday
11588	Christ College ATM	29/09/17	143	700400	130897756	Friday

11589 rows × 6 columns

Dapat dilihat untuk nama hari sudah menjadi sama dan siap untuk menjawab pertanyaan.

```python
hari_keseluruhan = atm.groupby('Weekday')['No Of Withdrawals'].sum()
hari_teramai = hari_keseluruhan.idxmax()
hari_tersepi = hari_keseluruhan.idxmin()

print(f'Sebaran pengunjung ATM:\n {hari_keseluruhan}\n')
print(f'ATM paling sering dikunjungi pada hari: {hari_teramai}\n')
print(f'ATM paling jarang dikunjungi pada hari: {hari_tersepi}')
```

```
Sebaran pengunjung ATM:
 Weekday
Friday       167463
Monday       181986
Saturday     224705
Sunday       311304
Thursday     174770
Tuesday      185715
Wednesday    183457
Name: No Of Withdrawals, dtype: int64

ATM paling sering dikunjungi pada hari: Sunday

ATM paling jarang dikunjungi pada hari: Friday
```

Kode di atas menggunakan metode groupby untuk menganalisis jumlah penarikan tunai ATM berdasarkan hari dalam seminggu. Hasilnya adalah daftar sebaran pengunjung ATM untuk setiap hari, serta mencatat hari yang paling sering dan paling jarang dikunjungi di ATM. Kode ini membantu dalam memahami pola aktivitas penggunaan ATM berdasarkan hari.

- Lakukan deteksi outlier untuk melihat potensi adanya fraud berdasarkan atribut Total amount Withdrawn (in Rupiah) pada dataset ATM Transaction menggunakan metode InterQuartileRange dan visualisasikan menggunakan BoxPlot serta sebutkan nama No Of Withdrawals dan nama ATM-nya yang terdeteksi fraud.

```python
# Konversi kolom "Total amount Withdrawn (Rupiah)" menjadi tipe data numerik
atm['Total amount Withdrawn (Rupiah)'] = pd.to_numeric(atm['Total amount Withdrawn (Rupiah)'], errors='coerce')

# Hitung IQR untuk kolom "Total amount Withdrawn (Rupiah)"
Q1 = atm['Total amount Withdrawn (Rupiah)'].quantile(0.25)
Q3 = atm['Total amount Withdrawn (Rupiah)'].quantile(0.75)
IQR = Q3 - Q1

# Tentukan batas bawah dan batas atas
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
```

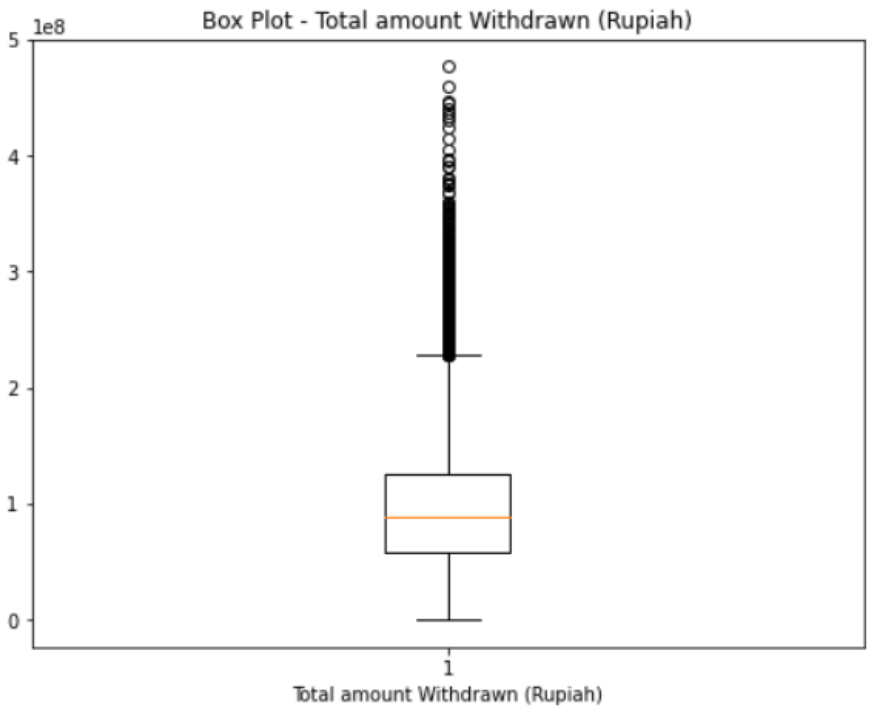
Kode diatas mengonversi kolom “Total amount Withdrawn (Rupiah)” ke tipe data numerik. Selanjutnya, Menghitung batas bawah dan batas atas menggunakan metode IQR (Interquartile Range) dengan mengambil Q1 dan Q3 dari data. Metode ini digunakan untuk mengidentifikasi outlier atau nilai-nilai ekstrem dalam suatu kolom dalam analisis data.

- Identifikasi dan Visualisasi Outlier dengan Box Plot

```python
# Identifikasi outlier
outliers = atm[(atm['Total amount Withdrawn (Rupiah)'] < lower_bound) | (atm['Total amount Withdrawn (Rupiah)'] > upper_b

# Visualisasikan dengan Box Plot
plt.figure(figsize=(8, 6))
plt.boxplot(atm['Total amount Withdrawn (Rupiah)'], vert=True)
plt.title('Box Plot - Total amount Withdrawn (Rupiah)')
plt.xlabel('Total amount Withdrawn (Rupiah)')
plt.show()
```

Output dari visualisasi tersebut:



Dari hasil visualisasi diatas, terdapat banyak titik lingkaran di atass garis boxplot, menandakan bahwa terdapat banyak potensi outlier di atas batas atas whisker. Outlier merupakan nilai-nilai yang jauh dari nilai lain dalam distribusi data.

- Memisahkan outlier dari DataFrame utama dan memberi nama variabel baru

```
fraud_atm = outliers[['No Of Withdrawals', 'ATM Name']]
fraud_atm
```

Beberapa output outlier dapat dilihat sebagai berikut:

	No Of Withdrawals	ATM Name			
13	309	KK Nagar ATM	**2616**	279	KK Nagar ATM
153	302	KK Nagar ATM	**2621**	372	KK Nagar ATM
338	286	KK Nagar ATM	**2626**	361	KK Nagar ATM
608	244	KK Nagar ATM	**2641**	300	KK Nagar ATM
758	252	KK Nagar ATM	**2651**	332	KK Nagar ATM
1691	251	KK Nagar ATM	**2730**	348	KK Nagar ATM
2441	261	KK Nagar ATM	**2740**	400	KK Nagar ATM
2576	306	KK Nagar ATM	**2745**	260	KK Nagar ATM
2581	353	KK Nagar ATM	**2750**	308	KK Nagar ATM
2586	261	KK Nagar ATM	**2755**	269	KK Nagar ATM
2591	300	KK Nagar ATM	**2760**	290	KK Nagar ATM
			2765	309	KK Nagar ATM