

Tugas Kecil 2 IF3170 Intelegensi Buatan
Exploratory Data Analysis



Oleh :

Akhmad Setiawan 13521164

Satria Octavianus Nababan 13521168

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2023

Jawaban Soal

Data yang digunakan dalam analisis adalah data latih berupa kumpulan dari beberapa aspek spesifikasi sebuah *handphone* terhadap kategori rentang harga *handphone* tersebut. Data yang diperoleh memiliki 1400 jumlah baris *record* data dan 21 atribut. *Overview* dari data tersebut diperoleh dengan melihat data secara singkat sebagai berikut.

```
# import library yang dibutuhkan
import pandas as pd
from IPython.display import display
import seaborn as sns
from matplotlib import pyplot as plt

# read csv
df = pd.read_csv("../data/data_train.csv")

target = df[["price_range"]]
nonTarget = df.drop(["price_range"], axis=1)
pd.set_option('display.max_columns', None)
print("Overview data:")
df
```

Dengan hasil *overview* data sebagai berikut.

Overview data:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
0	804	1	0.8	1	12	1	41	0.9	89	1	13	709	818	2027	11	5	11	1	0	0	1
1	1042	0	2.2	0	15	1	11	0.6	139	5	16	68	1018	2826	18	0	2	1	0	0	2
2	1481	1	2.0	1	0	0	35	0.5	105	3	0	249	522	2635	17	16	4	1	0	1	2
3	1104	0	1.7	0	1	1	60	0.4	199	2	13	653	1413	1229	6	0	3	1	1	1	0
4	652	0	0.5	1	1	0	58	0.6	142	3	2	464	781	565	18	12	9	0	0	1	0
...
1395	536	1	1.4	0	0	1	53	0.7	135	3	0	547	705	1211	15	10	7	1	0	1	0
1396	1097	0	0.8	0	10	1	21	0.1	160	7	15	1277	1352	2219	15	6	12	1	0	1	2
1397	1179	1	0.5	0	7	1	32	0.3	182	2	12	85	1451	340	16	5	16	1	0	0	0
1398	719	1	0.5	1	0	1	23	0.4	113	6	9	431	1727	3990	14	9	12	1	1	1	3
1399	1439	0	0.9	0	12	1	20	0.8	147	1	17	626	932	1790	19	12	15	1	0	1	1

1400 rows x 21 columns

Gambar 1 Overview Data

1. Statistik Dasar

Statistik dasar data yang diperiksa untuk setiap kolom data antara lain:

- mean* (rata-rata)
- median* (nilai tengah)
- STD (standar deviasi)
- minimum* (nilai terkecil pada kolom)
- 25% (kuartil 1)
- 50% (kuartil 2)
- 75% (kuartil 3)
- maximum* (nilai terbesar pada kolom)
- variance* (variansi)
- range* (jangkauan antar nilai minimal dan maksimal)
- IQR (*interquantil range* yakni jangkauan antara kuartil 1 dan kuartil 3)
- skewness* (kemiringan atau ukuran simetris data)
- kurtosis* (*tailedness* atau derajat keruncingan)
- mode* (modus yakni data yang paling sering muncul pada sebuah kolom)

Setiap statistik data mulai dari *mean* hingga *kurtosis* ditampilkan dalam sebuah tabel sebagai berikut.

	Descriptive Stat	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
0	Mean	1237.146	0.494	1.522	0.503	4.275	0.530	31.962	0.508	139.376	4.481	9.917	643.178	1251.717	2106.731	12.286	5.665	11.042	0.761	0.489	0.495	1.478
1	Median	1219.000	0.000	1.500	1.000	3.000	1.000	32.000	0.500	139.000	4.000	10.000	561.000	1247.000	2102.000	12.000	5.000	11.000	1.000	0.000	0.000	1.000
2	STD	430.052	0.500	0.815	0.500	4.324	0.499	18.163	0.289	35.401	2.280	6.080	444.629	428.983	1078.347	4.204	4.372	5.399	0.427	0.500	0.500	1.118
3	Minimum	501.000	0.000	0.500	0.000	0.000	0.000	2.000	0.100	80.000	1.000	0.000	0.000	500.000	256.000	5.000	0.000	2.000	0.000	0.000	0.000	0.000
4	25%	864.750	0.000	0.700	0.000	1.000	0.000	16.000	0.200	108.000	2.000	5.000	273.750	876.500	1201.000	9.000	2.000	6.000	1.000	0.000	0.000	0.000
5	50%	1219.000	0.000	1.500	1.000	3.000	1.000	32.000	0.500	139.000	4.000	10.000	561.000	1247.000	2102.000	12.000	5.000	11.000	1.000	0.000	0.000	1.000
6	75%	1602.000	1.000	2.200	1.000	7.000	1.000	48.000	0.800	169.000	7.000	15.000	950.250	1627.500	3035.750	16.000	9.000	16.000	1.000	1.000	1.000	2.000
7	Maximum	1998.000	1.000	3.000	1.000	19.000	1.000	64.000	1.000	200.000	8.000	20.000	1960.000	1998.000	3998.000	19.000	18.000	20.000	1.000	1.000	1.000	3.000
8	Variance	184944.538	0.250	0.664	0.250	18.698	0.249	329.893	0.083	1253.217	5.198	36.967	197694.930	184026.286	1162832.850	17.675	19.116	29.150	0.182	0.250	0.250	1.249
9	Range	1497.000	1.000	2.500	1.000	19.000	1.000	62.000	0.900	120.000	7.000	20.000	1960.000	1498.000	3742.000	14.000	18.000	18.000	1.000	1.000	1.000	3.000
10	IQR	737.250	1.000	1.500	1.000	6.000	1.000	32.000	0.600	61.000	5.000	10.000	676.500	751.000	1834.750	7.000	7.000	10.000	0.000	1.000	1.000	2.000
11	Skewness	0.042	0.026	0.166	-0.011	1.020	-0.120	0.063	0.059	0.020	0.020	0.029	0.659	0.004	0.029	-0.103	0.671	-0.009	-1.223	0.043	0.020	0.029
12	Kurtosis	-1.168	-2.002	-1.330	-2.003	0.293	-1.988	-1.227	-1.267	-1.210	-1.232	-1.164	-0.316	-1.176	-1.186	-1.183	-0.335	-1.192	-0.504	-2.001	-2.002	-1.358

Gambar 1.1 Statistik *Mean* hingga *Kurtosis*

Sementara itu, modus ditampilkan dalam tabel terpisah karena pada kolom tertentu terdapat beberapa nilai modus (beberapa nilai pada kolom tersebut memiliki frekuensi kemunculan yang sama dan merupakan frekuensi kemunculan terbesar).

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
0	772	0.0	0.5	1.0	0.0	1.0	27.0	0.1	182.0	4.0	10.0	88.0	1247.0	1229.0	17.0	1.0	15.0	1.0	0.0	0.0	0.0
1	1068	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	347.0	NaN	3142.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	1330	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	526.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	1872	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	1949	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Gambar 1.2 Modus Setiap Kolom

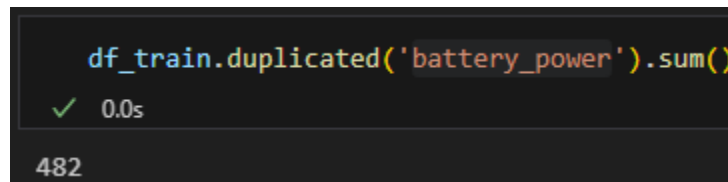
Dapat dilihat pada **Gambar 1.2** terdapat beberapa kolom yang memiliki modus lebih dari satu nilai. Kolom *battery_power* memiliki 5 nilai modus, kolom *px_height* memiliki tiga nilai modus, dan kolom *ram* memiliki 2 nilai modus.

2. *Duplicate Value*

Duplicate value, atau nilai ganda, merujuk pada situasi di mana ada dua atau lebih entri dalam dataset yang memiliki nilai yang sama atau identik pada semua atribut yang diamati. Dalam konteks dataset, nilai ganda dapat muncul jika dua atau lebih observasi berbagi semua nilai yang sama untuk semua atribut atau kolom yang ada.

Dilakukan pengecekan *duplicate value* pada setiap kolom dari `data_train.csv` :

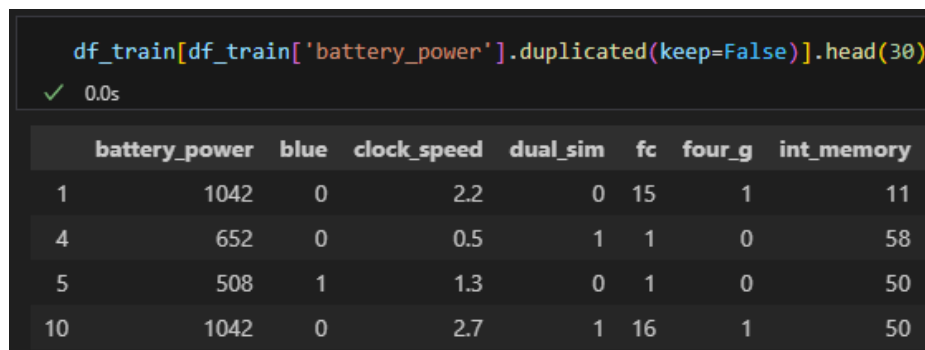
1. Terdapat 482 *duplicate value* pada kolom `battery_power`



```
df_train.duplicated('battery_power').sum()
✓ 0.0s
482
```

Gambar 2.1 Jumlah Data Duplikat pada Kolom *battery_power*

Dalam konteks ini, *battery_power* adalah kolom yang mengukur total energi baterai dalam satu waktu, diukur dalam milliampere-jam (mAh). Ini mengindikasikan bahwa terdapat 482 ponsel atau perangkat yang memiliki nilai kapasitas baterai yang identik atau sama satu dengan yang lainnya.

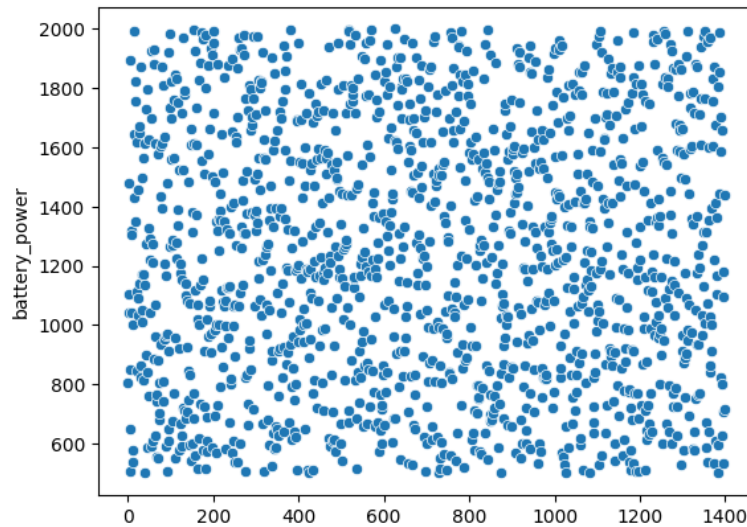


```
df_train[df_train['battery_power'].duplicated(keep=False)].head(30)
✓ 0.0s
```

	<code>battery_power</code>	<code>blue</code>	<code>clock_speed</code>	<code>dual_sim</code>	<code>fc</code>	<code>four_g</code>	<code>int_memory</code>
1	1042	0	2.2	0	15	1	11
4	652	0	0.5	1	1	0	58
5	508	1	1.3	0	1	0	50
10	1042	0	2.7	1	16	1	50

Gambar 2.2 Contoh Duplikasi Data pada Kolom *battery_power*

Dari gambar diatas dapat dilihat salah satu contoh data kolom *battery_power* yang memiliki nilai yang sama yaitu pada baris 1 dan 10.



Gambar 2.3 Sebaran Data *battery_power*

Dari visualisasi *scatterplot* diatas dapat dilihat bahwa kolom *battery_power* memiliki nilai yang sangat variatif.

2. Terdapat beberapa kolom yang hanya memiliki 2 variasi data yaitu kolom *blue*, *dual_sim*, *four_g*, *three_g*, *touch_screen*, *wifi*. Hal ini karena kolom-kolom tersebut hanya menginformasikan nilai 1 (*true*) dan 0 (*false*). Dengan sedikitnya variasi nilai data pada kolom-kolom tersebut maka sangat wajar jika semua barisnya merupakan *duplicate value*.

```
df_train.duplicated('blue').sum()
✓ 0.0s
1398
```

```
df_train.duplicated('dual_sim').sum()
✓ 0.0s
1398
```

```
df_train.duplicated('four_g').sum()
✓ 0.0s
1398
```

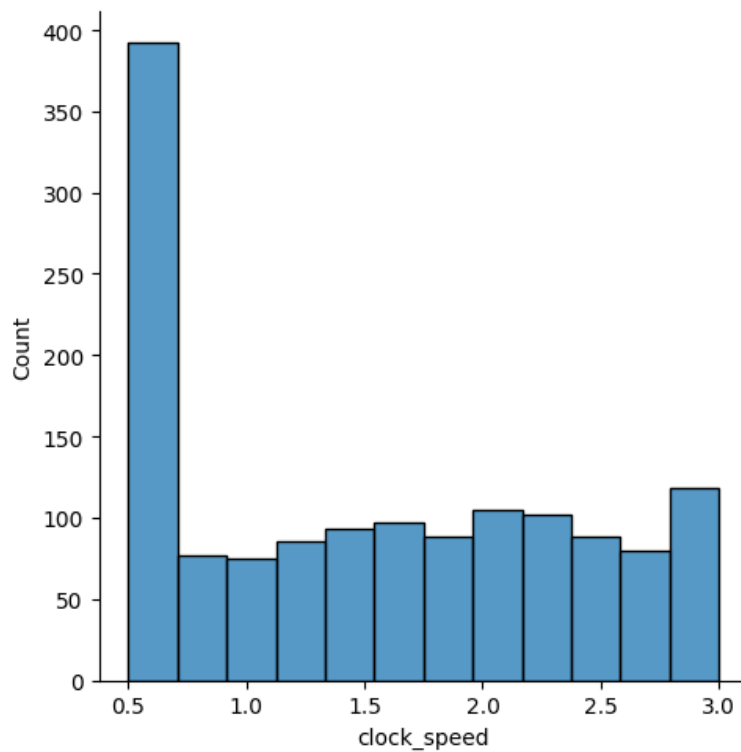
```
df_train.duplicated('three_g').sum()
✓ 0.0s
1398
```

```
df_train.duplicated('touch_screen').sum()
✓ 0.0s
1398
```

```
df_train.duplicated('wifi').sum()
✓ 0.0s
1398
```

Gambar 2.4 Kasus Duplikasi Nilai pada Kolom Non Numerik

3. Pada kolom *clock_speed* persebaran datanya berada pada rentang nilai 0.5 - 3.0.



Gambar 2.5 Sebaran Data *clock_speed*

Terdapat sejumlah 1374 *duplicate value* kecepatan mikroprosesor dalam menjalankan instruksi.

```
df_train.duplicated('clock_speed').sum()
✓ 0.0s
1374
```

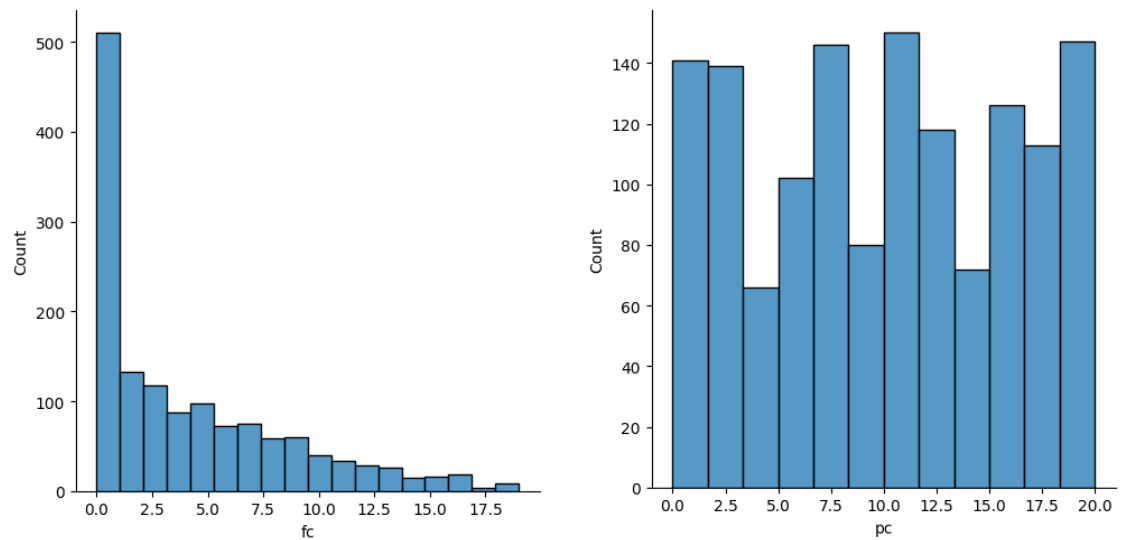
Gambar 2.6 Jumlah Data Duplikat pada Kolom *clock_speed*

4. Kolom *fc* menginformasikan resolusi kamera depan sedangkan kolom *pc* menginformasikan resolusi kamera utama perangkat dalam megapiksel. Pada kolom *fc* terdapat sejumlah 1380 *duplicate value* tetapi dengan nilai yang cukup variatif pada *range* 0-17.5 sedangkan pada kolom *pc* terdapat 1379 *duplicate value* dengan sebaran nilai pada *range* 0-20.

```
df_train.duplicated('fc').sum()
✓ 0.0s
1380
```

```
df_train.duplicated('pc').sum()
✓ 0.0s
1379
```

Gambar 2.7 Jumlah Data Duplikat pada Kolom *fc* dan *pc*

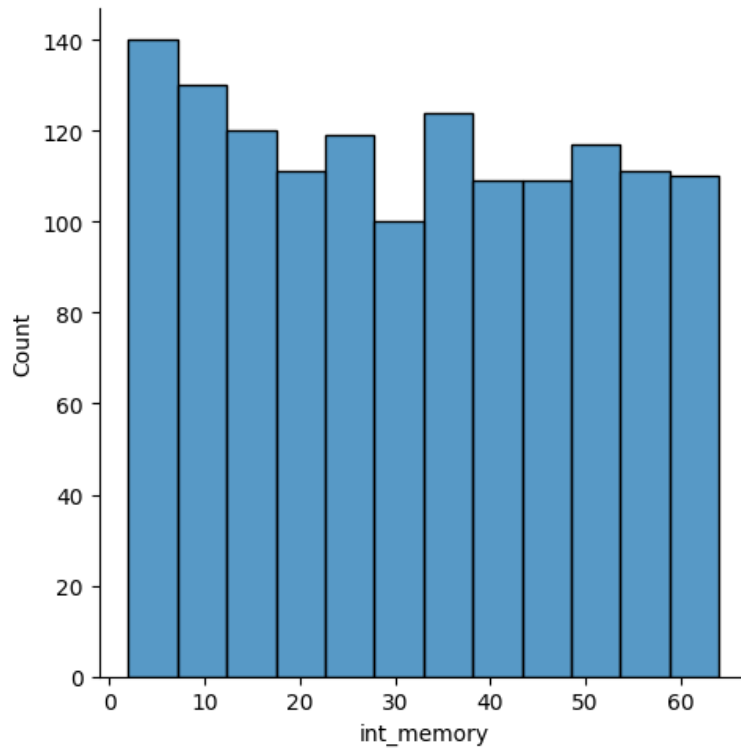


Gambar 2.8 Sebaran Data pada Kolom *fc* dan *pc*

5. Kolom *int_memory* menginformasikan kapasitas memori internal dalam gigabyte, terdapat sejumlah 1337 *duplicate value*.

```
df_train.duplicated('int_memory').sum()
✓ 0.0s
1337
```

Gambar 2.9 Jumlah Data Duplikat pada Kolom *int_memory*



Gambar 2.10 Sebaran Data pada Kolom *int_memory*

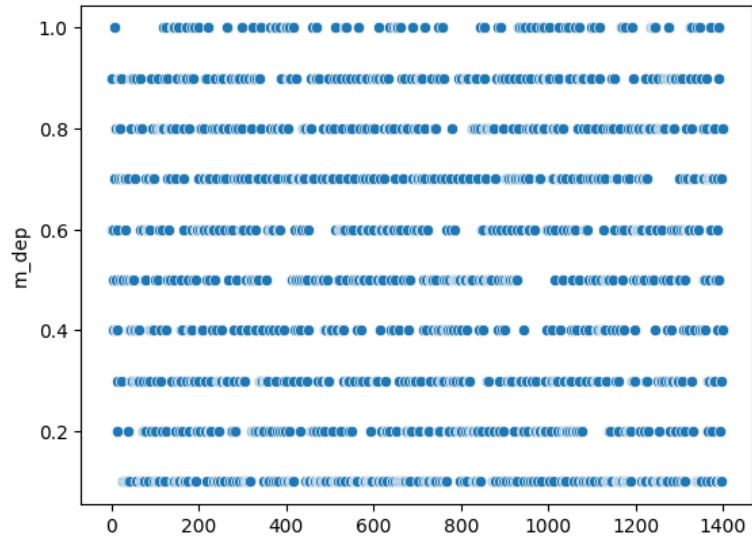
Dari gambar diatas, dapat dilihat bahwa *duplicate value* pada kolom *int_memory* memiliki persebaran jumlah yang cukup merata.

6. Kolom *m_dep* menginformasikan ketebalan ponsel dalam cm, yang mana nilainya hanya berada pada rentang 0-1 saja, sehingga sangat besar kemungkinan terjadi *duplicate value*.

```
df_train.duplicated('m_dep').sum()
✓ 0.0s
1390
```

Gambar 2.11 Jumlah Data Duplikat pada Kolom *m_dep*

Terdapat sejumlah 1390 *duplicate value* dengan persebaran data yang cukup konsisten.

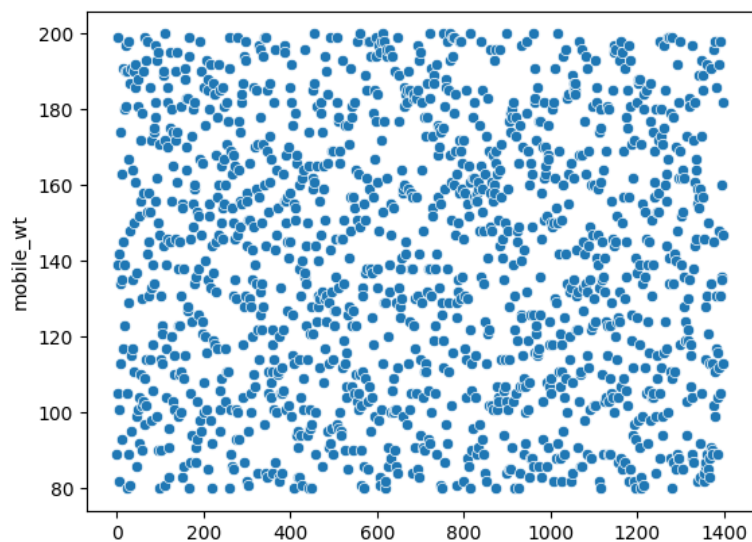


Gambar 2.12 Sebaran Data pada Kolom *m_dep*

7. Berat ponsel direpresentasikan oleh kolom *mobile_wt* yang memiliki sejumlah 1279 *duplicate value*.

```
df_train.duplicated('mobile_wt').sum()
✓ 0.0s
1279
```

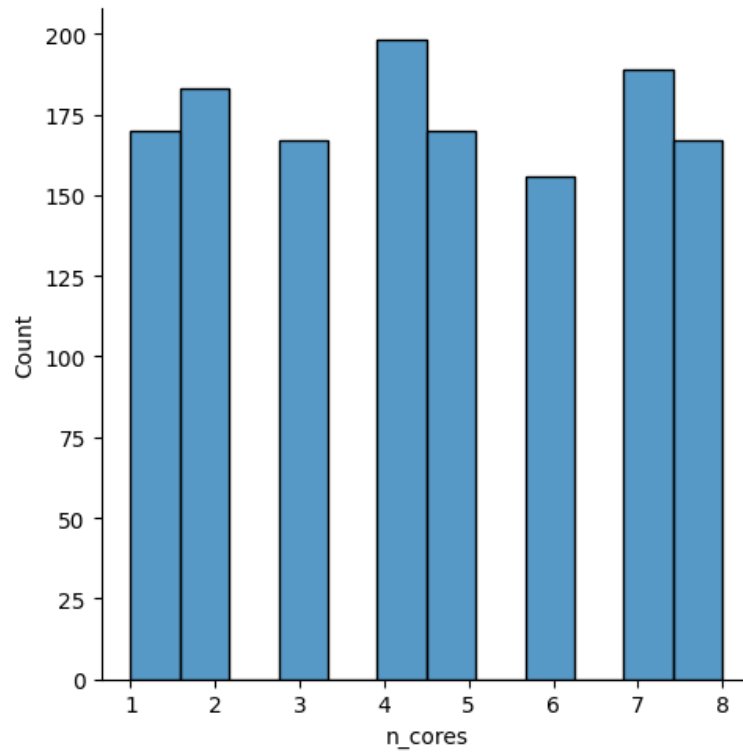
Gambar 2.13 Jumlah Data Duplikat pada Kolom *mobile_wt*



Gambar 2.14 Sebaran Data pada Kolom *mobile_wt*

Persebaran nilainya cukup variatif yang terdapat pada *range* 80-200.

8. Kolom *n_cores* menunjukkan jumlah *core prosesor* dengan persebaran *nilai* yang cukup konsisten pada range 1-7.



Gambar 2.15 Sebaran Data pada Kolom *n_cores*

Kolom ini terdapat sejumlah 1392 *duplicate value*.

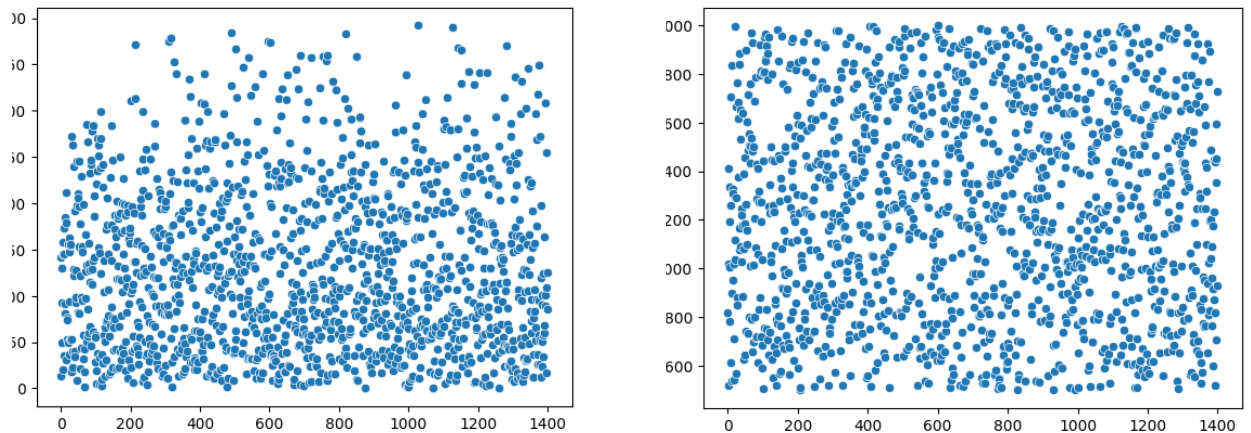
```
df_train.duplicated('n_cores').sum()
✓ 0.0s
1392
```

Gambar 2.16 Jumlah Data Duplikat pada Kolom *n_cores*

9. Kolom *px_height* dan *px_width* menunjukkan tinggi dan lebar resolusi piksel. Masing-masing memiliki sejumlah 465 dan 492 *duplicate value*.

```
df_train.duplicated('px_height').sum()
✓ 0.0s
465
```

```
df_train.duplicated('px_width').sum()
✓ 0.0s
492
```



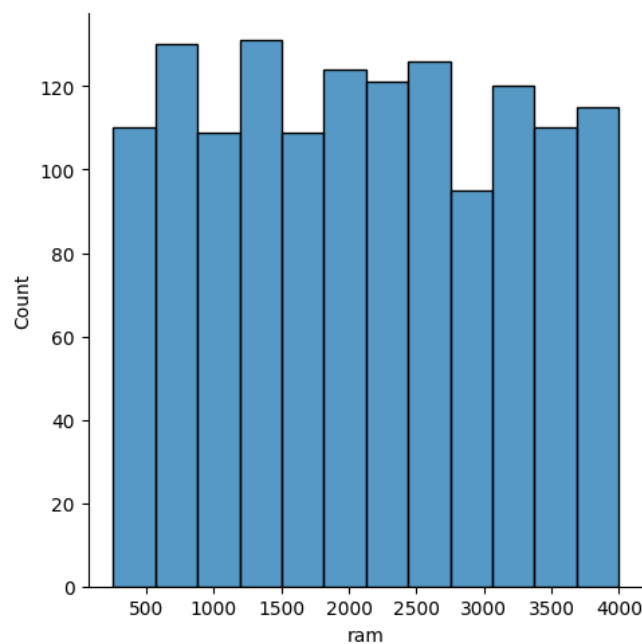
Gambar 2.17 Jumlah Duplikat dan Sebaran Data Kolom *px_height* dan *px_width*

Terlihat bahwa persebaran nilai dari tabel *px_height* dan *px_width* cukup variatif.

10. Kolom *ram* menunjukkan ukuran RAM dalam *megabyte* yang memiliki sejumlah 230 *duplicate value*.

```
df_train.duplicated('ram').sum()
✓ 0.0s
230
```

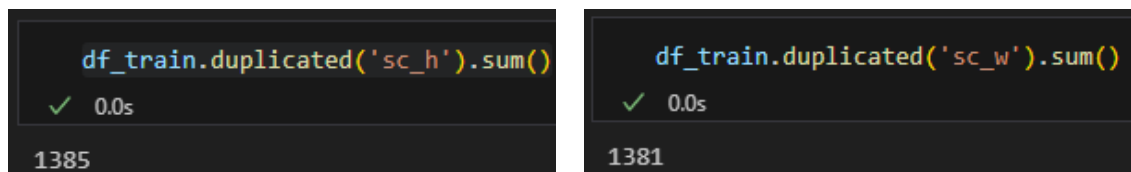
Gambar 2.18 Jumlah Data Duplikat pada Kolom *ram*



Gambar 2.19 Sebaran Data pada Kolom *ram*

Dapat dilihat bahwa persebaran ukuran RAM cukup merata pada *range* jumlah 90-130.

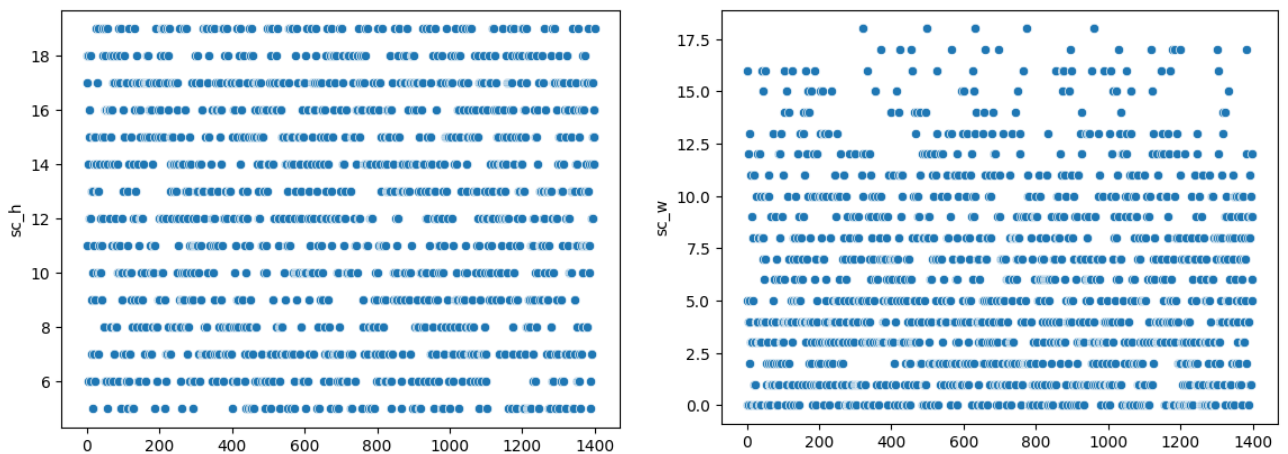
11. Kolom *sc_h* dan *sc_w* masing-masing menunjukkan tinggi dan lebar layar ponsel dalam cm. Kolom *sc_h* memiliki total 1385 *duplicate value* dan *sc_w* memiliki 1381 *duplicate value*.



```
df_train.duplicated('sc_h').sum()
✓ 0.0s
1385
```

```
df_train.duplicated('sc_w').sum()
✓ 0.0s
1381
```

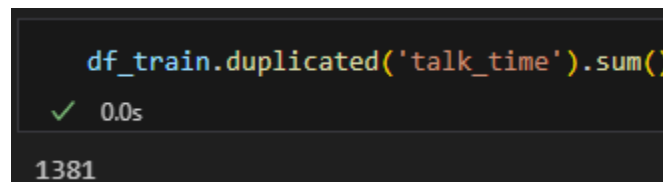
Gambar 2.20 Jumlah Data Duplikat pada Kolom *sc_h* dan *sc_w*



Gambar 2.21 Sebaran Data pada Kolom *sc_h* dan *sc_w*

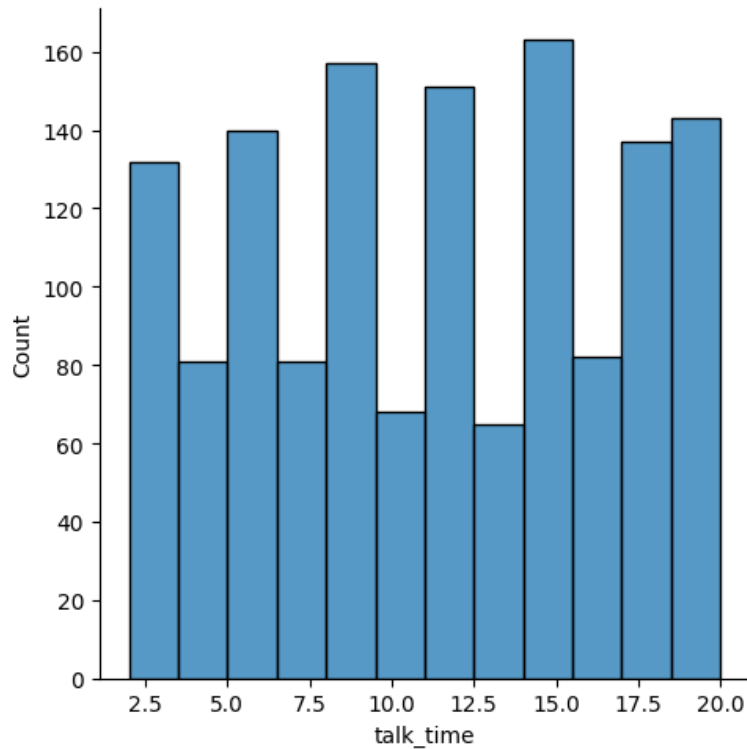
Dapat dilihat bahwa persebaran data *sc_h* dan *sc_w* cukup konsisten.

12. Waktu telepon maksimum dalam satu kali pengisian baterai ditunjukkan oleh kolom *talk_time* yang memiliki sejumlah 1381 *duplicate value*.



```
df_train.duplicated('talk_time').sum()
✓ 0.0s
1381
```

Gambar 2.22 Jumlah Data Duplikat pada Kolom *talk_time*



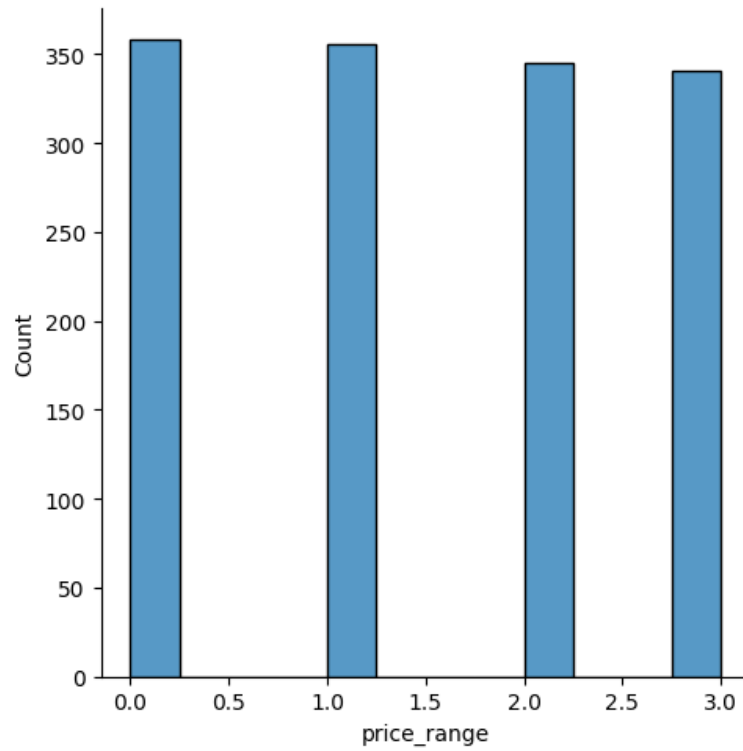
Gambar 2.23 Sebaran Data pada Kolom *talk_time*

Waktu telepon maksimum dalam satu kali pengisian baterai berada pada range 2.5-20.0 jam dengan persebaran waktu yang cukup konsisten.

- Kolom *price_range* memiliki sejumlah 1396 *duplicate value* karena hanya terdapat rentang harga dengan nilai 0 (biaya rendah), 1 (biaya sedang), 2 (biaya tinggi) atau 3 (biaya sangat tinggi) sehingga kemungkinan terjadinya *duplicate value* sangat besar.

```
df_train.duplicated('price_range').sum()
✓ 0.0s
1396
```

Gambar 2.24 Jumlah Data Duplikat pada Kolom *price_range*



Gambar 2.25 Sebaran Data pada Kolom *price_range*

Dapat dilihat bahwa nilai rentang harga ponsel tersebar cukup merata pada setiap harganya.

3. *Missing Value*

Missing value merujuk pada situasi di mana data atau informasi yang seharusnya ada untuk suatu variabel atau atribut dalam dataset tetapi tidak ada atau tidak tersedia. Ini berarti bahwa ada beberapa observasi atau entri yang tidak memiliki nilai atau informasi yang *valid* untuk atribut tersebut.

Pada `data_train.csv` tidak ditemukan adanya *missing value*, klaim ini dapat dilihat dari hasil pemeriksaan berikut ini.

```
df_train.isnull().sum()
✓ 0.0s

battery_power    0
blue             0
clock_speed      0
dual_sim         0
fc              0
four_g           0
int_memory       0
m_dep            0
mobile_wt        0
n_cores          0
pc               0
px_height        0
px_width         0
ram              0
sc_h             0
sc_w             0
talk_time        0
three_g          0
touch_screen     0
wifi             0
price_range      0
dtype: int64
```

Gambar 3.1 Justifikasi *Missing Value*

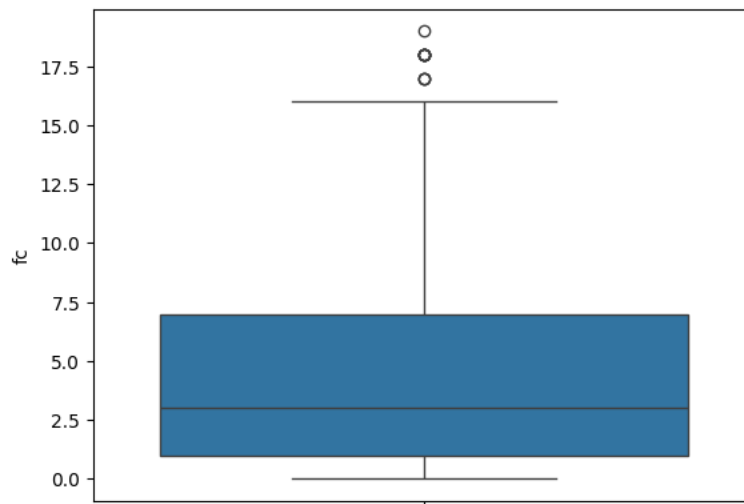
Pada kode tersebut seluruh kolom pada `data_train.csv` diperiksa jumlah kemungkinan *missing value* yang ada, dan dapat dilihat bahwa dari semua kolom yang diperiksa bahwa jumlah *missing value* tidak ada atau 0.

4. *Outliers*

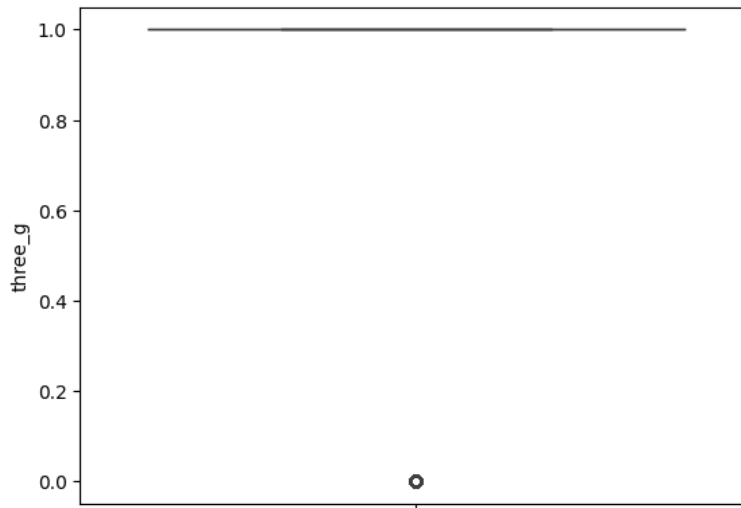
Outlier merujuk kepada data yang secara signifikan berbeda dari nilai-nilai lain dalam himpunan data. *Outlier* dapat menjadi nilai yang sangat ekstrem, terlalu tinggi atau terlalu rendah dibandingkan dengan sebagian besar data lainnya. *Outlier* bisa menjadi hasil dari kesalahan pengukuran, gangguan dalam pengumpulan data, atau mungkin saja merupakan data yang valid namun sangat tidak umum.

Berikut adalah langkah-langkah analisis data *outlier* pada data latih menggunakan metode *interquartile range* (IQR).

1. Untuk menentukan *outlier*, awalnya kami melakukan visualisasi menggunakan grafik *boxplot* terhadap seluruh kolom data_train.csv. Dari hasil visualisasi menggunakan grafik *boxplot* terhadap seluruh kolom, dapat terlihat bahwa pada kolom **fc** dan kolom **three_g** terdapat *outlier* berupa persebaran data yang berbeda dari nilai-nilai lain dalam himpunan data.



Gambar 4.1 Visualisasi *Boxplot* Awal Pada Kolom **fc**



Gambar 4.2 Visualisasi *Boxplot* Awal Pada Kolom *three_g*

2. Untuk semakin meyakinkan hasil pengecekan yang dilakukan, dilakukan pencarian terhadap nilai *upper limit* dan *lower limit* dari kolom *fc* dan *three_g*. *Upper limit* dan *lower limit* adalah nilai tertentu yang digunakan untuk menentukan batas maksimum dan batas minimum yang diperbolehkan atau yang masih dapat diterima dalam suatu konteks. Dalam statistik, *upper limit* dan *lower limit* dapat digunakan dalam deteksi *outlier*, yaitu nilai di atas *upper limit* dan dibawah *lower limit* dianggap sebagai *outlier*.
3. Untuk menentukan *upper limit* dan *lower limit*, awalnya dicari terlebih dahulu nilai dari kuartil 1 dan kuartil 3. Dari selisih kuartil 1 dan kuartil 3 dapat ditentukan nilai dari *interquartile range* (dapat dilihat juga pada rangkuman statistik pada nomor 1).

```
q1 = df_train['fc'].quantile(0.25)
q3 = df_train['fc'].quantile(0.75)
iqr = q3-q1
✓ 0.0s
```

```
q1, q3, iqr
✓ 0.0s
(1.0, 7.0, 6.0)
```

Gambar 4.3 Pencarian Q1, Q3, dan IQR pada Kolom *fc*

Didapat nilai dari kuartil 1 = 1.0, kuartil 3 = 7.0, dan IQR = 6.0

4. Kemudian *upper limit* dan *lower limit* dihitung menggunakan rumus berikut,

```

upper_limit = q3 + (1.5 * iqr)
lower_limit = q1 - (1.5 * iqr)
lower_limit, upper_limit
✓ 0.0s
(-8.0, 16.0)

```

Gambar 4.3 Pencarian *Upper Limit* dan *Lower Limit* Kolom *fc*

- Untuk memperbaiki persebaran data maka *outlier* kami hapus menggunakan 2 metode yaitu *percentile trimming* dan *capping*. Prinsip utama dari metode *trimming* adalah menghilangkan sebagian data dengan nilai ekstrem, yang dianggap sebagai gangguan, sehingga analisis statistik lebih stabil. *Percentile trimming* akan menghapus nilai yang berada di luar batasan tertentu dalam distribusi persentil tertentu.

```

## trimming - delete the outlier data
new_df = df_train.loc[(df_train['fc'] <= upper_limit) & (df_train['fc'] >= lower_limit)]
print('before removing outlier : ', len(df_train))
print('after removing outlier : ', len(new_df))
print('outlier : ', len(df_train)-len(new_df))
✓ 0.0s

before removing outlier : 1400
after removing outlier : 1389
outlier : 11

```

Gambar 4.4 Metode *Trimming Outliers* Kolom *fc*

Metode *trimming* dapat mengubah karakteristik distribusi data, seperti rata-rata dan deviasi standar. Oleh karena itu, penggunaan metode *trimming* harus dilakukan dengan hati-hati dan mempertimbangkan tujuan analisis.

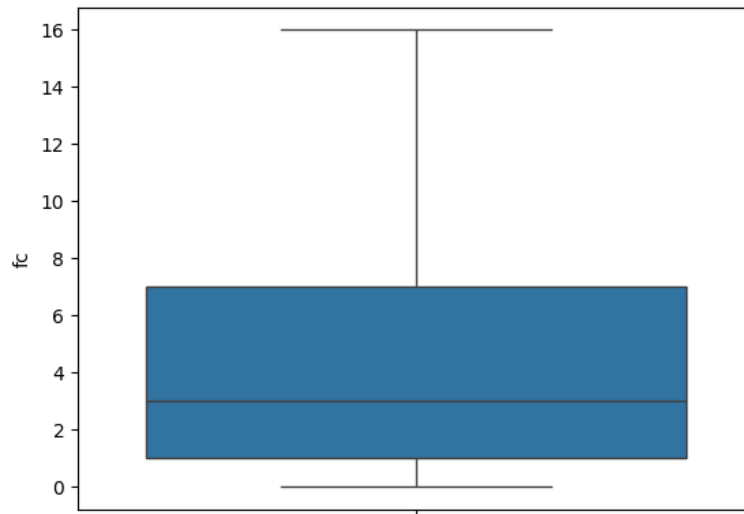
- Visualisasi dari *data_train.csv* setelah *outliers* dari kolom *fc* dihapus menggunakan metode *trimming*.

```

# Plot setelah outlier dihapus menggunakan metode trimming
sns.boxplot(new_df['fc'])
✓ 0.3s

<Axes: ylabel='fc'>

```



Gambar 4.5 Visualisasi Kolom *fc* Setelah *Outliers* Dihapus Dengan *Trimming*

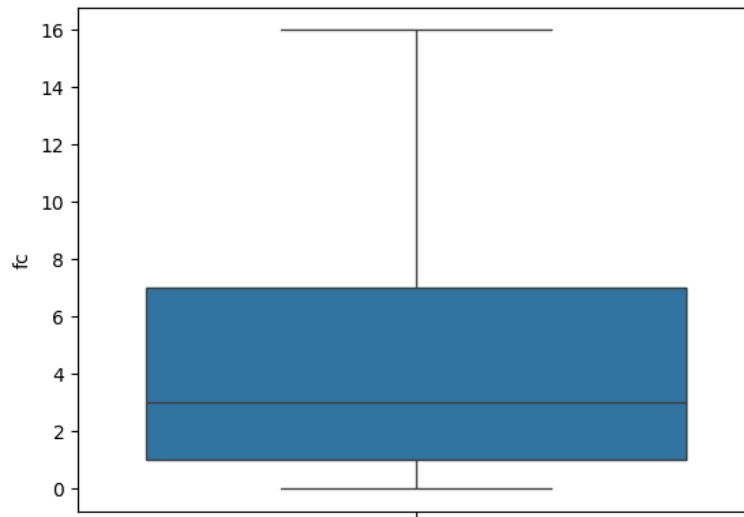
7. Selain metode *trimming*, metode *capping* juga digunakan untuk menghapus *outlier*. Metode *capping* melibatkan pembatasan atau pengubahan nilai *outlier* sehingga mereka tidak melebihi atau kurang dari batasan tertentu yang telah ditetapkan. Metode *capping* sering digunakan untuk membuat *outlier* lebih konsisten dengan data lainnya tanpa menghapusnya sepenuhnya.

```
# capping - change the outlier values to upper (or) lower limit values
new_df = df_train.copy()
new_df.loc[(new_df['fc'] >= upper_limit), 'fc'] = upper_limit
new_df.loc[(new_df['fc'] <= lower_limit), 'fc'] = lower_limit
✓ 0.0s
```

Gambar 4.6 Metode *Capping* pada *Outliers* Kolom *fc*

8. Visualisasi dari *data_train.csv* setelah *outliers* dari kolom *fc* dihapus menggunakan metode *capping*.

```
# Plot setelah outlier dihapus menggunakan metode capping
sns.boxplot(new_df['fc'])
✓ 0.3s
<Axes: ylabel='fc'>
```



Gambar 4.7 Visualisasi Kolom *fc* Setelah Dilakukan *Capping* untuk *Outliers*

9. Menggunakan metode dan langkah-langkah yang sama seperti sebelumnya, maka kita juga akan memperbaiki data *outliers* dari kolom *three_g*. Awalnya kita menentukan kuartil 1 dan kuartil 3. Dari selisih kuartil 1 dan kuartil 3 dapat ditentukan nilai dari *interquartile range* dari kolom *three_g*.

```
q1 = df_train['three_g'].quantile(0.25)
q3 = df_train['three_g'].quantile(0.75)
iqr = q3-q1
✓ 0.0s
```

```
q1, q3, iqr
✓ 0.0s
```

```
(1.0, 1.0, 0.0)
```

Gambar 4.8 Pencarian Q1, Q3, dan IQR pada Kolom *three_g*

10. Menentukan *upper limit* dan *lower limit* dari kolom *three_g*.

```
upper_limit = q3 + (1.5 * iqr)
lower_limit = q1 - (1.5 * iqr)
lower_limit, upper_limit
✓ 0.0s
```

```
(1.0, 1.0)
```

Gambar 4.9 Pencarian *Upper Limit* dan *Lower Limit* Kolom *three_g*

11. Menghapus data *outliers* pada kolom *three_g* menggunakan metode *trimming*.

```
## trimming - delete the outlier data
new_df = df_train.loc[(df_train['three_g'] <= upper_limit) & (df_train['three_g'] >= lower_limit)]
print('before removing outlier : ', len(df_train))
print('after removing outlier : ', len(new_df))
print('outliner : ', len(df_train)-len(new_df))

✓ 0.0s

before removing outlier : 1400
after removing outlier : 1065
outliner : 335
```

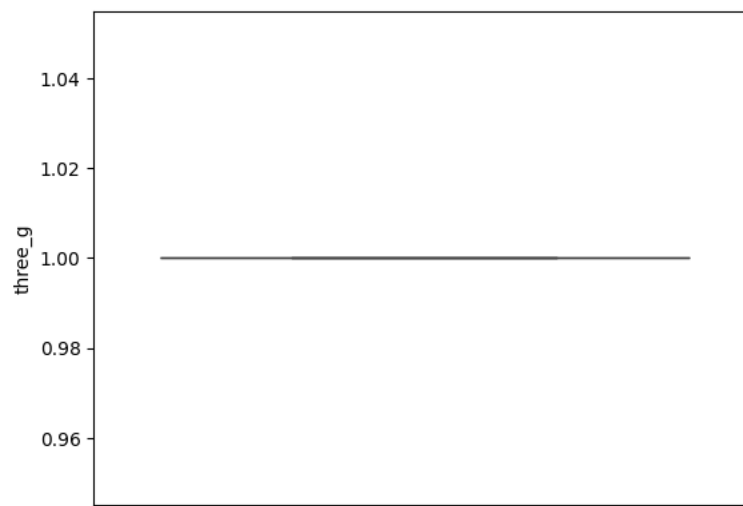
Gambar 4.10 Metode *Trimming Outliers* Kolom *three_g*

12. Visualisasi dari *data_train.csv* setelah *outliers* dari kolom *three_g* dihapus menggunakan metode *trimming*.

```
# Plot setelah outlier dihapus menggunakan metode trimming
sns.boxplot(new_df['three_g'])

✓ 0.2s

<Axes: ylabel='three_g'>
```



Gambar 4.11 Visualisasi Kolom *three_g* Setelah *Outliers* Dihapus Dengan *Trimming*

13. Menghapus data *outliers* pada kolom *three_g* menggunakan metode *capping*.

```
# capping - change the outlier values to upper (or) lower limit values
new_df = df_train.copy()
new_df.loc[(new_df['three_g'] >= upper_limit), 'three_g'] = upper_limit
new_df.loc[(new_df['three_g'] <= lower_limit), 'three_g'] = lower_limit
```

✓ 0.0s

Gambar 4.12 Metode *Capping* pada *Outliers* Kolom *three_g*

14. Visualisasi dari *data_train.csv* setelah *outliers* dari kolom *three_g* dihapus menggunakan metode *capping*.

```
# Plot setelah outlier dihapus menggunakan metode capping
sns.boxplot(new_df['three_g'])
```

✓ 0.3s

<Axes: ylabel='three_g'>



Gambar 4.13 Visualisasi Kolom *three_g* Setelah Dilakukan *Capping* untuk *Outliers*

5. Plot dan Analisis Kurtosis Data

Dalam visualisasi plot data yang digunakan, akan digunakan plot dan analisis kurtosis pada kolom numerik dan *bar chart* untuk kolom non numerik.

Berikut adalah metode yang digunakan dalam analisis plot pada kolom-kolom bertipe numerik:

```
# list kolom numerik
numeric =
["battery_power", "clock_speed", "fc", "int_memory", "m_dep", "mobile_wt", "n_
cores", "pc", "px_height", "px_width", "ram", "sc_h", "sc_w", "talk_time"]
# plotting kolom numerik
for i, column in enumerate(numeric, 1):
    print("Plot kurtosis untuk kolom", column)
    plt.figure(figsize=(6, 6))
    sns.histplot(df[column], kde=True)
    plt.title(f'{column} Kurtosis: {df[column].kurtosis():.3f}')
    plt.tight_layout()
    plt.show()
```

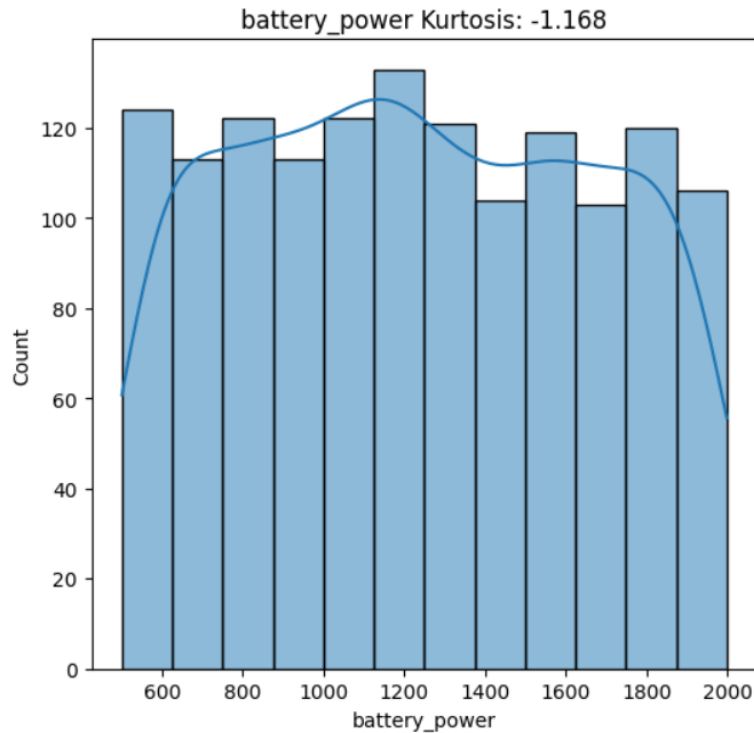

Berikut adalah metode yang digunakan dalam analisis *bar chart* pada kolom-kolom bertipe non numerik.

```
# list kolom non numerik
non_numeric =
["blue", "dual_sim", "four_g", "three_g", "touch_screen", "wifi", "price_range"]

# plotting kolom non numerik
for i, column in enumerate(non_numeric, 1):
    print("Bar Chart untuk Kolom", column)
    plt.figure(figsize=(6, 6))
    sns.countplot(data=df, x=column)
    plt.title(f'Bar Chart untuk {column}')
    ax = sns.countplot(data=df, x=column)
    for p in ax.patches:
        ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width()
/ 2, p.get_height()),
                    ha='center', va='center', xytext=(0, 10),
textcoords='offset points')
    plt.tight_layout()
    plt.show()
```

Analisis untuk setiap kolom-kolom data adalah sebagai berikut.

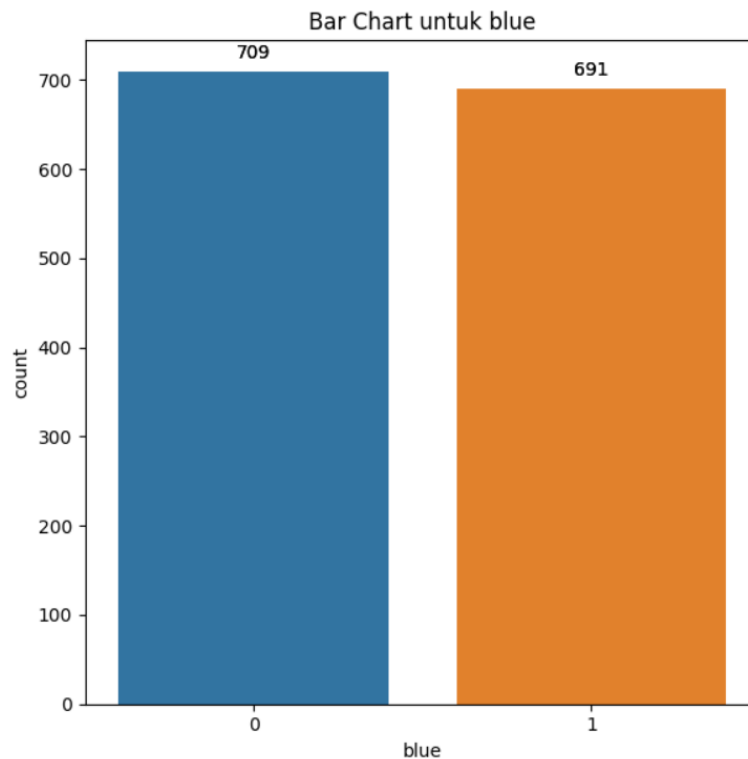
1. *battery_power*: Total energi baterai dalam satu waktu diukur dalam mAh (numerik)



Gambar 5.1 Sebaran Data Kolom *battery_power*

Terlihat pada **Gambar 5.1** di atas nilai kurtosis untuk data kolom *battery_power* bernilai -1.168 (bernilai negatif) sehingga tergolong ke dalam *platykurtic distribution*. Artinya, nilai pada kolom *battery_power* cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu). Hal ini dapat dijustifikasi dengan melihat persebaran data pada gambar yang cenderung sama.

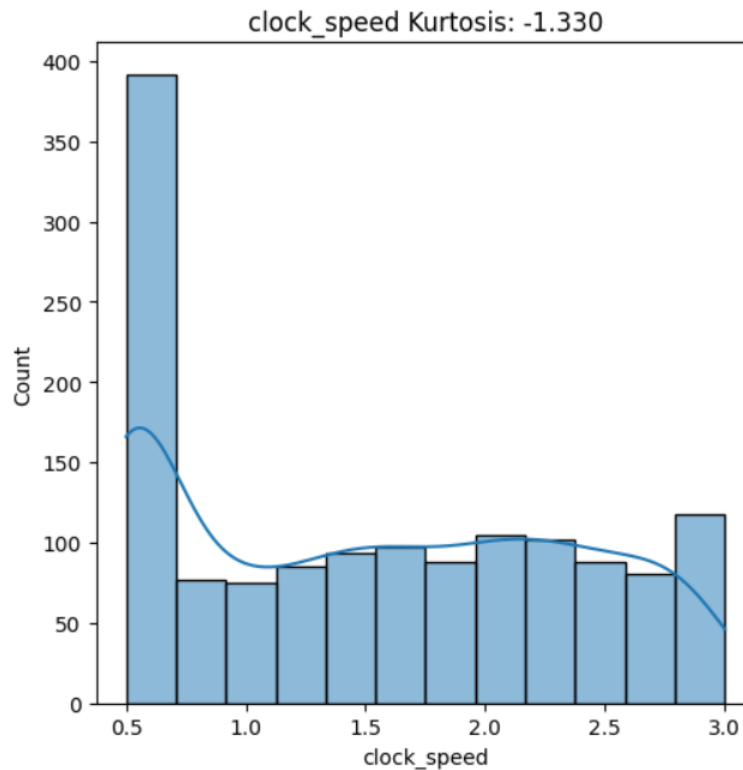
2. *blue*: Memiliki *bluetooth* atau tidak (non numerik)



Gambar 5.2 Perbandingan Jumlah Nilai Kolom *blue*

Kolom *blue* tergolong ke dalam kolom non numerik dengan nilai 1 = *true* dan 0 = *false*. Perbandingan frekuensi kedua nilai dapat dilihat pada **Gambar 5.2** di atas. Frekuensi nilai 0 sedikit lebih banyak dibandingkan frekuensi nilai 1. Artinya, ponsel yang tidak memiliki *bluetooth* pada data latih sedikit lebih banyak dibanding dengan ponsel yang memiliki *bluetooth*.

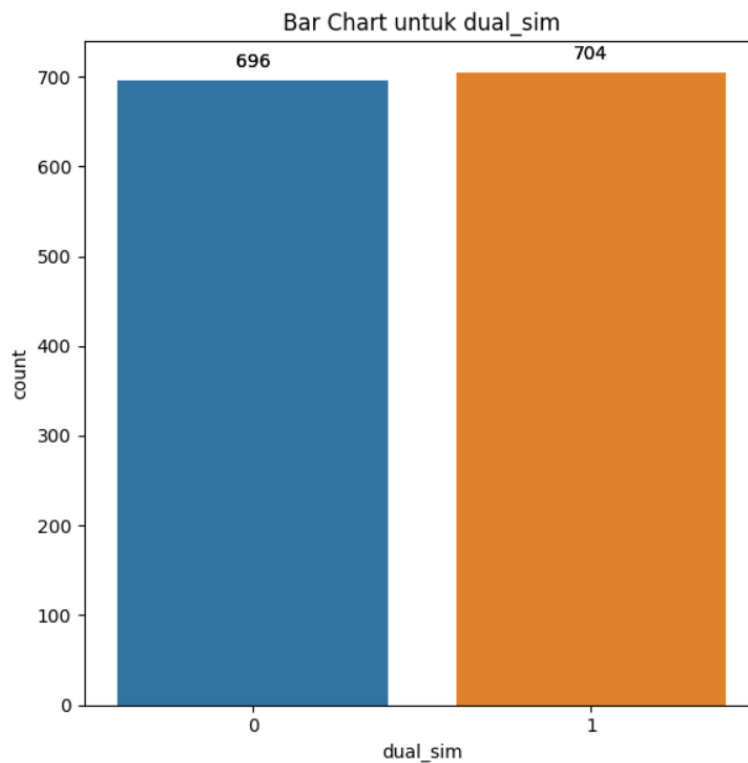
3. *clock_speed*: Kecepatan mikroprosesor menjalankan instruksi (numerik)



Gambar 5.3 Sebaran Data Kolom *clock_speed*

Terlihat pada **Gambar 5.3** di atas nilai kurtosis untuk data kolom *clock_speed* bernilai -1.330 (bernilai negatif) sehingga tergolong ke dalam *platykurtic distribution*. Artinya, nilai pada kolom *clock_speed* cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu). Meskipun pada grafik terlihat ada lonjakan jumlah data pada rentang 0.5 - 0.75, hal ini tidak begitu ekstrem yang berpengaruh terhadap nilai kurtosisnya karena diimbangi oleh sebaran nilai pada data di bagian kanannya.

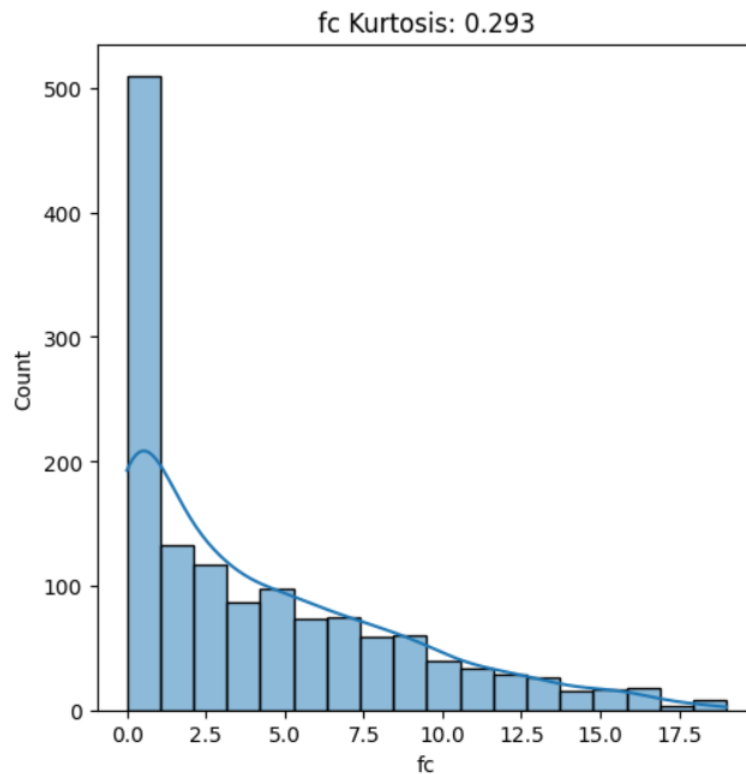
4. *dual_sim*: Memiliki dukungan dual sim atau tidak (non numerik)



Gambar 5.4 Perbandingan Jumlah Nilai Kolom *dual_sim*

Kolom *dual_sim* tergolong ke dalam kolom non numerik dengan nilai 1 = *true* dan 0 = *false*. Perbandingan frekuensi kedua nilai dapat dilihat pada **Gambar 5.4** di atas. Frekuensi nilai 1 sedikit lebih banyak dibandingkan frekuensi nilai 0. Artinya, ponsel yang mendukung *dual sim* pada data latih sedikit lebih banyak dibanding dengan ponsel yang tidak mendukung *dual sim*.

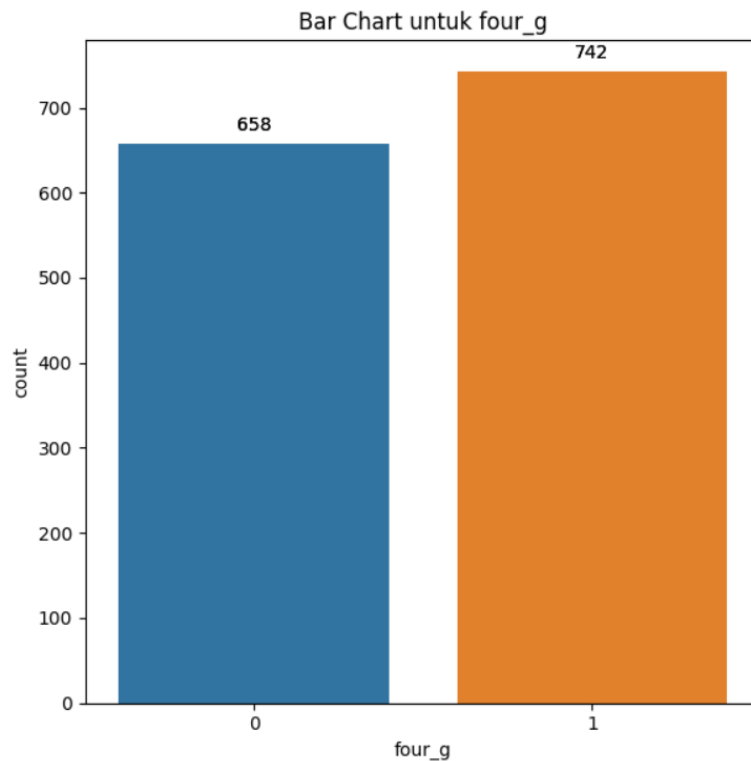
5. *fc*: Resolusi kamera depan dalam megapiksel (numerik)



Gambar 5.5 Sebaran Data Kolom *fc*

Terlihat pada **Gambar 5.5** di atas nilai kurtosis untuk data kolom *fc* bernilai 0.293 (bernilai positif) sehingga tergolong ke dalam *leptokurtic distribution*. Artinya, nilai pada kolom *fc* ada sebuah lonjakan data yang tidak seragam dengan data lainnya (berat pada satu titik tertentu). Pada gambar terlihat nilai pada rentang di sebelah kiri (rentang 0 - 1) memiliki jumlah frekuensi yang sangat tinggi, sementara nilai pada rentang di sebelah kanan semakin menurun. Hal inilah yang memengaruhi nilai kurtosis pada kolom *fc*.

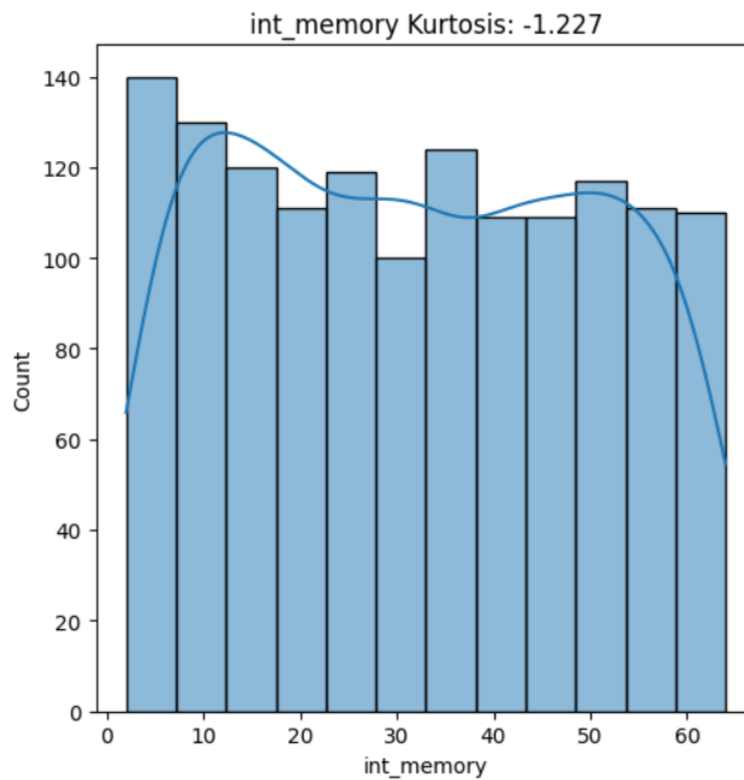
6. *four_g*: Memiliki 4G atau tidak (non numerik)



Gambar 5.6 Perbandingan Jumlah Nilai Kolom *four_g*

Kolom *four_g* tergolong ke dalam kolom non numerik dengan nilai 1 = *true* dan 0 = *false*. Perbandingan frekuensi kedua nilai dapat dilihat pada **Gambar 5.6** di atas. Frekuensi nilai 1 lebih banyak dibandingkan frekuensi nilai 0. Artinya, ponsel yang mendukung 4G pada data latih lebih banyak dibanding dengan ponsel yang tidak mendukung 4G.

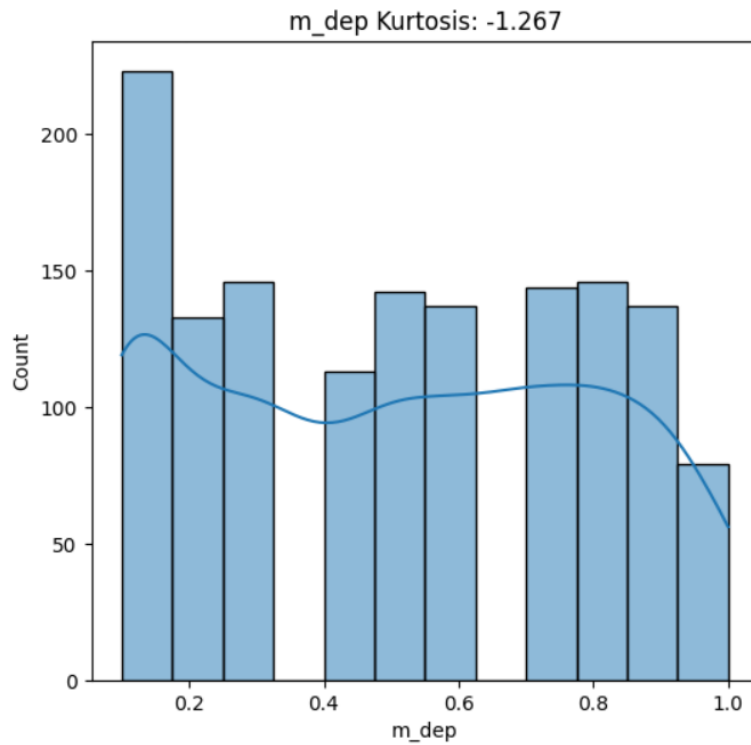
7. *int_memory*: Memori internal dalam *gigabyte* (numerik)



Gambar 5.7 Sebaran Data Kolom *int_memory*

Terlihat pada **Gambar 5.7** di atas nilai kurtosis untuk data kolom *int_memory* bernilai -1.227 (bernilai negatif) sehingga tergolong ke dalam *platykurtic distribution*. Artinya, nilai pada kolom *int_memory* cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu). Hal ini dapat dijustifikasi dengan melihat persebaran data pada gambar yang cenderung sama.

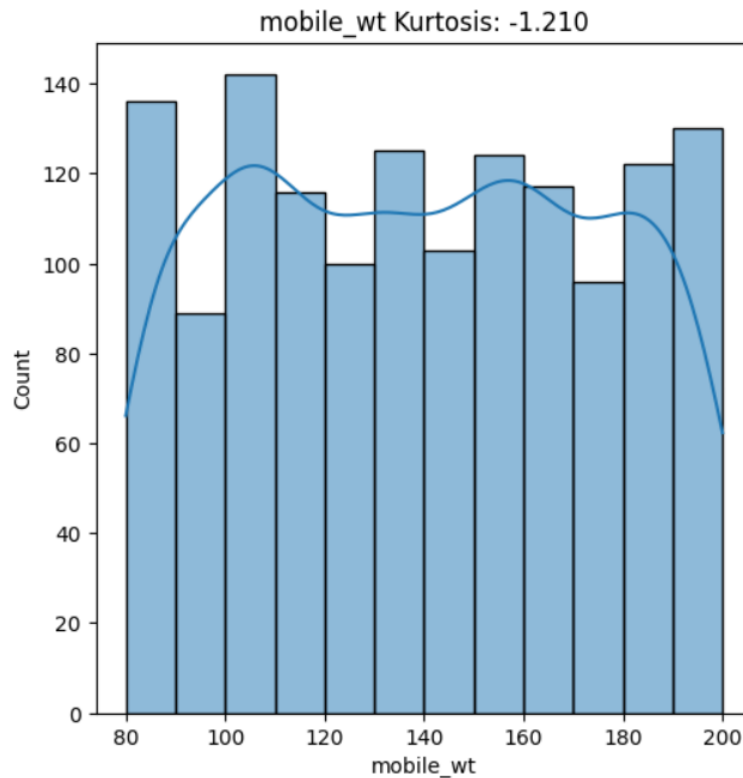
8. m_dep : Ketebalan ponsel dalam cm (numerik)



Gambar 5.8 Sebaran Data Kolom m_dep

Terlihat pada **Gambar 5.8** di atas nilai kurtosis untuk data kolom m_dep bernilai -1.267 (bernilai negatif) sehingga tergolong ke dalam *platykurtic distribution*. Artinya, nilai pada kolom m_dep cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu). Meskipun pada grafik terlihat ada lonjakan jumlah data pada rentang 0.1 - 0.2, hal ini tidak begitu ekstrem yang berpengaruh terhadap nilai kurtosisnya karena diimbangi oleh sebaran nilai pada data di bagian kanannya.

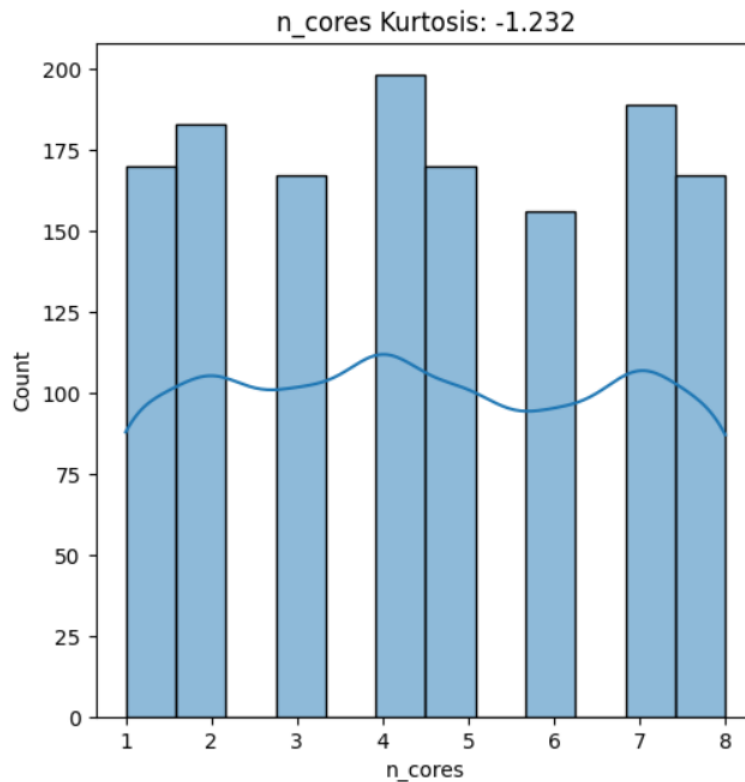
9. *mobile_wt*: Berat ponsel (numerik)



Gambar 5.9 Sebaran Data Kolom *mobile_wt*

Terlihat pada **Gambar 5.9** di atas nilai kurtosis untuk data kolom *mobile_wt* bernilai -1.210 (bernilai negatif) sehingga tergolong ke dalam *platykurtic distribution*. Artinya, nilai pada kolom *mobile_wt* cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu). Hal ini dapat dijustifikasi dengan melihat persebaran data pada gambar yang cenderung sama.

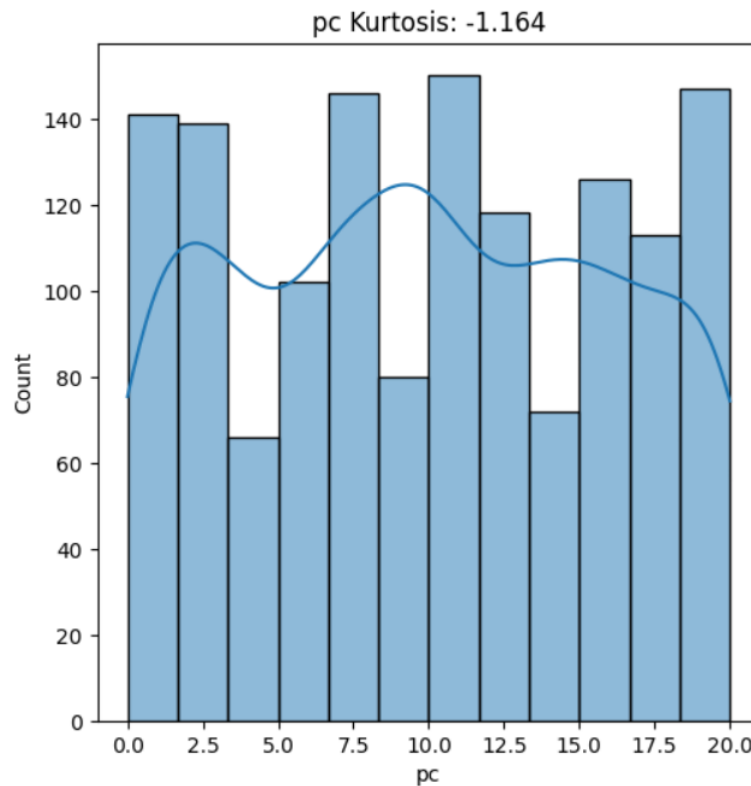
10. n_cores : Jumlah *core* prosesor (numerik)



Gambar 5.10 Sebaran Data Kolom n_cores

Terlihat pada **Gambar 5.10** di atas nilai kurtosis untuk data kolom n_cores bernilai -1.232 (bernilai negatif) sehingga tergolong ke dalam *platykurtic distribution*. Artinya, nilai pada kolom n_cores cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu). Hal ini dapat dijustifikasi dengan melihat persebaran data pada gambar yang cenderung sama.

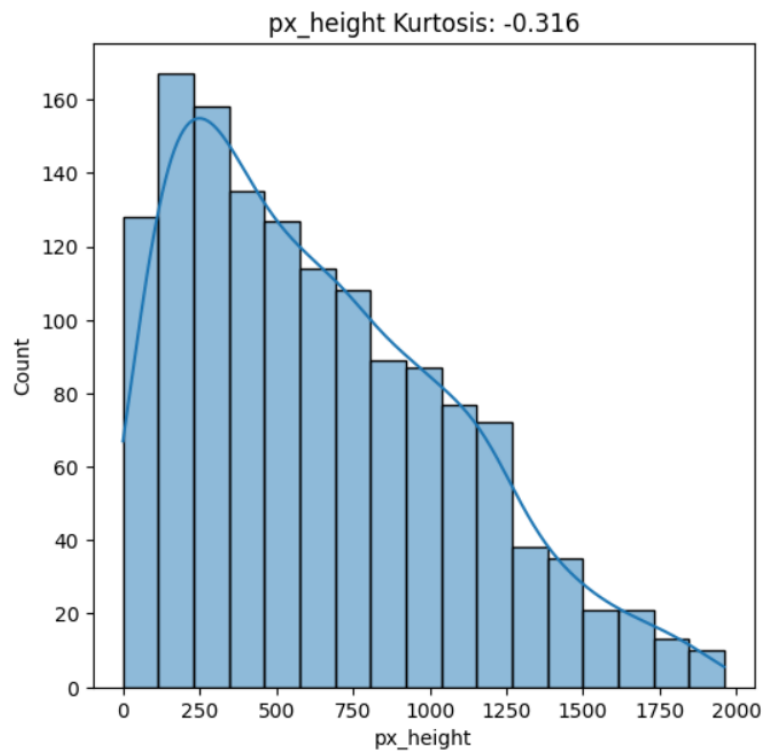
11. *pc*: Resolusi kamera utama dalam megapiksel (numerik)



Gambar 5.11 Sebaran Data Kolom *pc*

Terlihat pada **Gambar 5.11** di atas nilai kurtosis untuk data kolom *pc* bernilai -1.164 (bernilai negatif) sehingga tergolong ke dalam *platykurtic distribution*. Artinya, nilai pada kolom *pc* cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu). Hal ini dapat dijustifikasi dengan melihat persebaran data pada gambar yang cenderung sama.

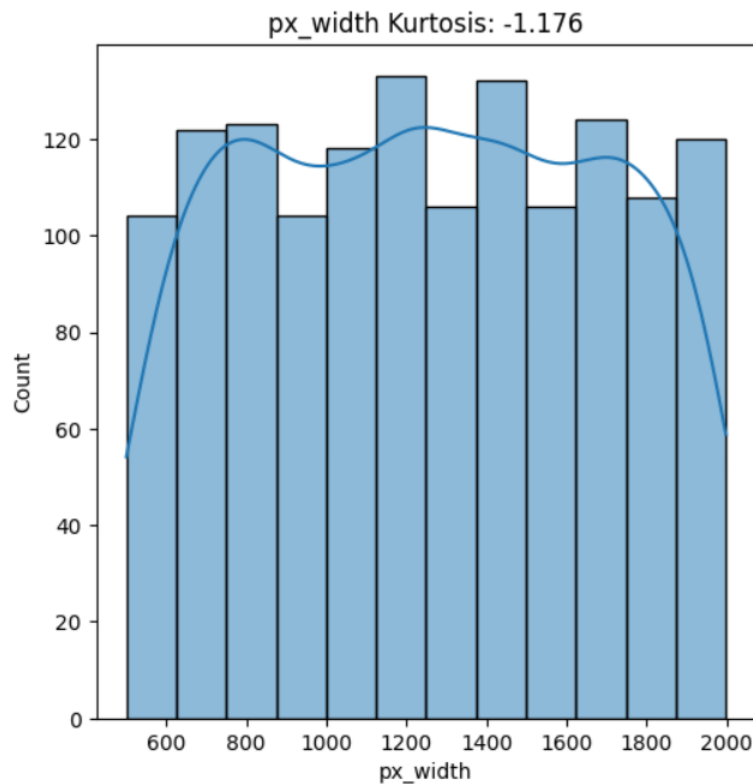
12. *px_height*: Tinggi resolusi piksel (numerik)



Gambar 5.12 Sebaran Data Kolom *px_height*

Terlihat pada **Gambar 5.12** di atas nilai kurtosis untuk data kolom *px_height* bernilai -0.316 (bernilai negatif mendekati 0) sehingga masih tergolong ke dalam *platykurtic distribution*. Jika merujuk pada gambar, terlihat sebaran data sedikit melonjak pada rentang sekitar 250 dan menurun setelahnya. Jika dilihat pada nilai kurtosisnya yang negatif, sebaran data pada kolom *px_height* masih cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu).

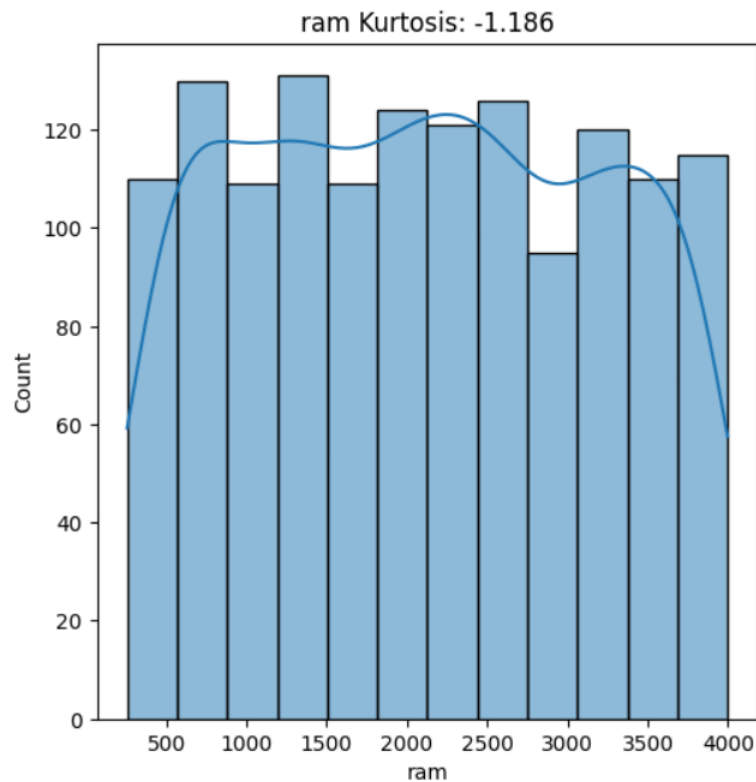
13. *px_width*: Lebar resolusi piksel (numerik)



Gambar 5.13 Sebaran Data Kolom *px_width*

Terlihat pada **Gambar 5.13** di atas nilai kurtosis untuk data kolom *px_width* bernilai -1.176 (bernilai negatif) sehingga tergolong ke dalam *platykurtic distribution*. Artinya, nilai pada kolom *px_width* cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu). Hal ini dapat dijustifikasi dengan melihat persebaran data pada gambar yang cenderung sama.

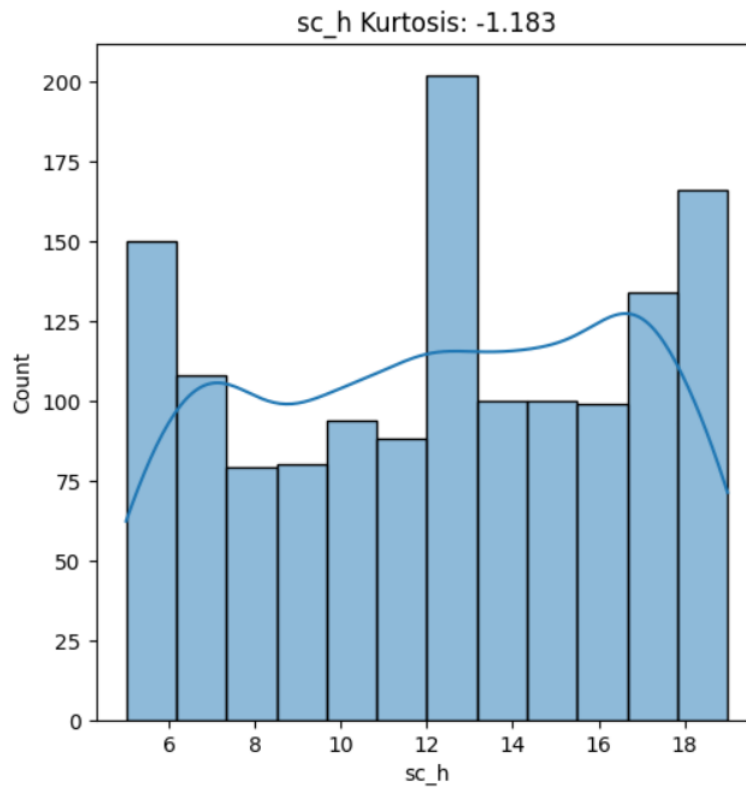
14. *ram*: Ukuran RAM dalam *megabyte* (numerik)



Gambar 5.14 Sebaran Data Kolom *ram*

Terlihat pada **Gambar 5.14** di atas nilai kurtosis untuk data kolom *ram* bernilai -1.186 (bernilai negatif) sehingga tergolong ke dalam *platykurtic distribution*. Artinya, nilai pada kolom *ram* cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu). Hal ini dapat dijustifikasi dengan melihat persebaran data pada gambar yang cenderung sama.

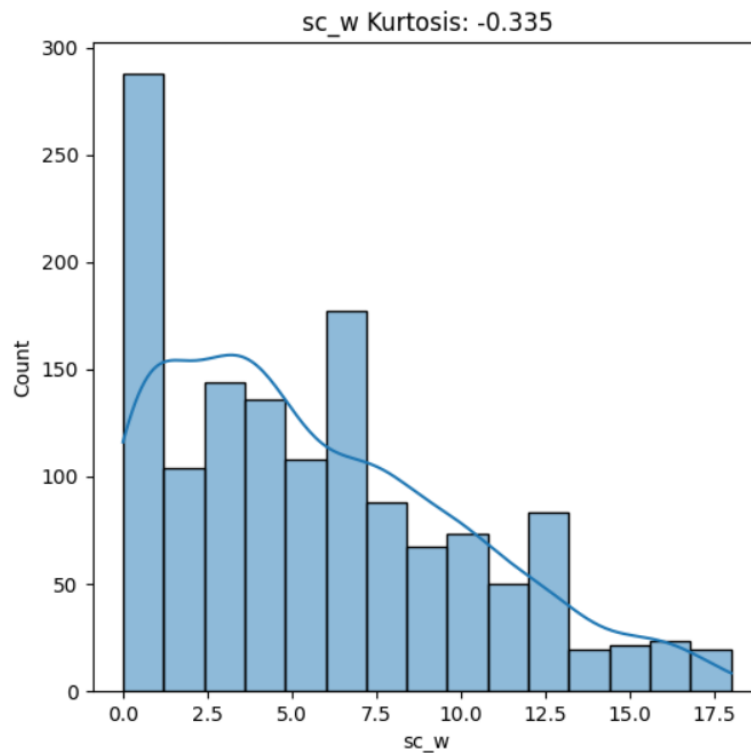
15. sc_h : Tinggi layar ponsel dalam cm (numerik)



Gambar 5.15 Sebaran Data Kolom sc_h

Terlihat pada **Gambar 5.15** di atas nilai kurtosis untuk data kolom sc_h bernilai -1.183 (bernilai negatif) sehingga tergolong ke dalam *platykurtic distribution*. Artinya, nilai pada kolom sc_h cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu). Hal ini dapat dijustifikasi dengan melihat persebaran data pada gambar yang cenderung sama.

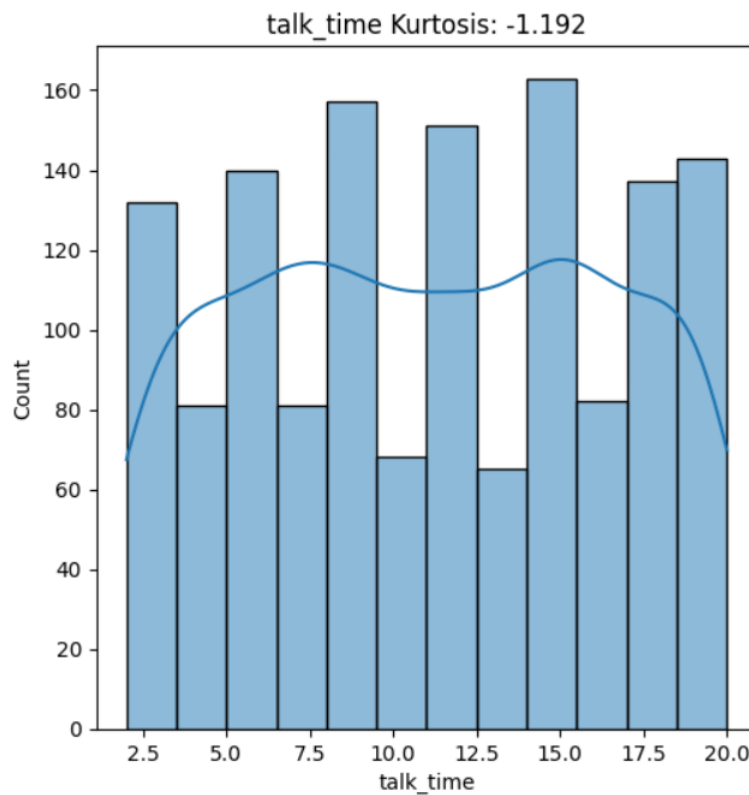
16. `sc_w`: Lebar layar ponsel dalam cm (numerik)



Gambar 5.16 Sebaran Data Kolom `sc_w`

Terlihat pada **Gambar 5.16** di atas nilai kurtosis untuk data kolom `sc_w` bernilai -0.335 (bernilai negatif mendekati 0) sehingga masih tergolong ke dalam *platykurtic distribution*. Jika merujuk pada gambar, terlihat sebaran data sedikit melonjak pada rentang 0 - 1 dan cenderung menurun setelahnya. Jika dilihat pada nilai kurtosisnya yang negatif, sebaran data pada kolom `sc_w` masih cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu).

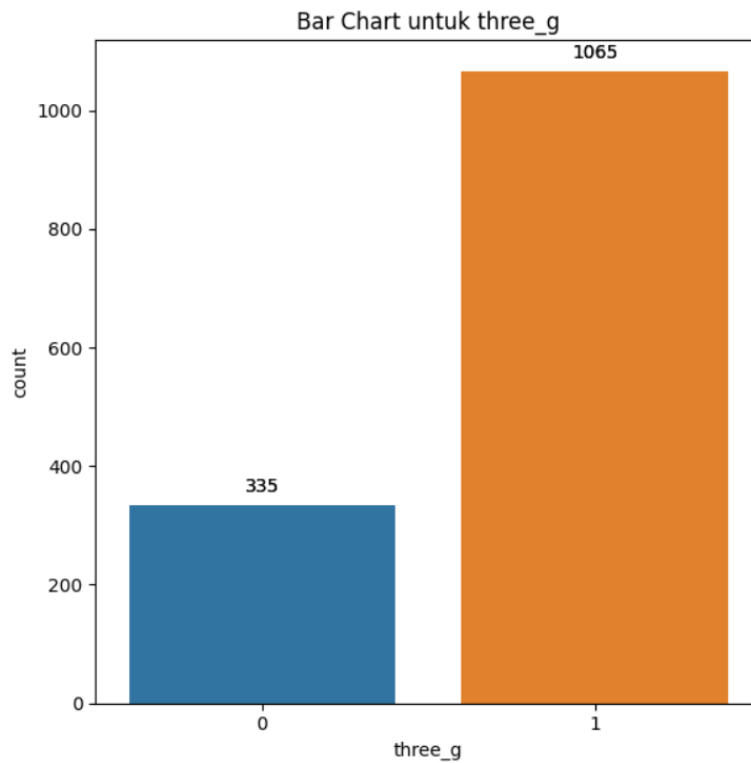
17. *talk_time*: Waktu telepon maksimum dalam satu kali pengisian baterai (numerik)



Gambar 5.17 Sebaran Data Kolom *talk_time*

Terlihat pada **Gambar 5.17** di atas nilai kurtosis untuk data kolom *talk_time* bernilai -1.192 (bernilai negatif) sehingga tergolong ke dalam *platykurtic distribution*. Artinya, nilai pada kolom *talk_time* cenderung terdistribusi secara merata (tidak bertumpu pada satu titik tertentu). Hal ini dapat dijustifikasi dengan melihat persebaran data pada gambar yang cenderung sama.

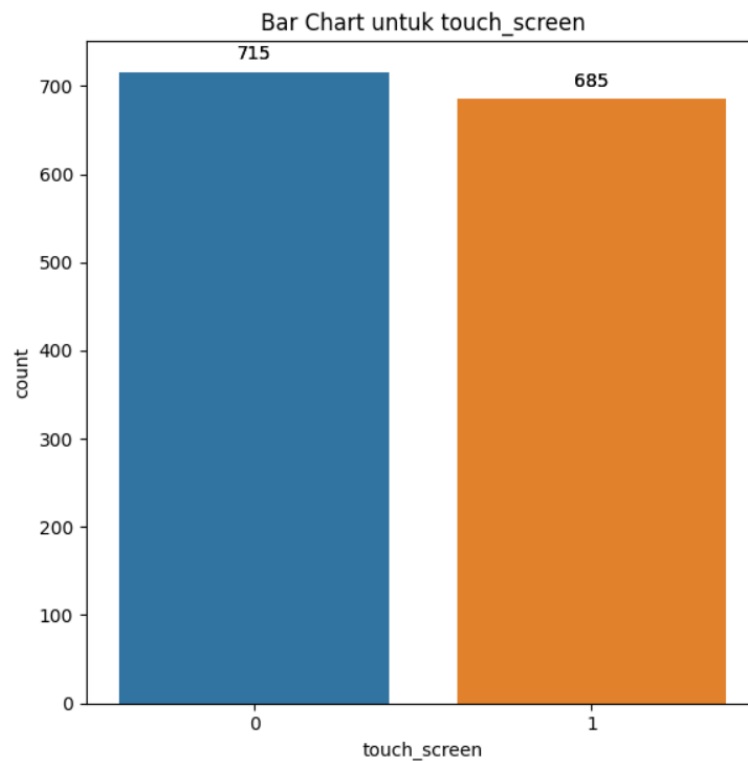
18. *three_g*: Memiliki 3G atau tidak (non numerik)



Gambar 5.18 Perbandingan Jumlah Nilai Kolom *three_g*

Kolom *three_g* tergolong ke dalam kolom non numerik dengan nilai 1 = *true* dan 0 = *false*. Perbandingan frekuensi kedua nilai dapat dilihat pada **Gambar 5.18** di atas. Frekuensi nilai 1 jauh lebih banyak dibandingkan frekuensi nilai 0. Artinya, ponsel yang mendukung 3G pada data latih juga jauh lebih banyak dibanding dengan ponsel yang tidak mendukung 3G.

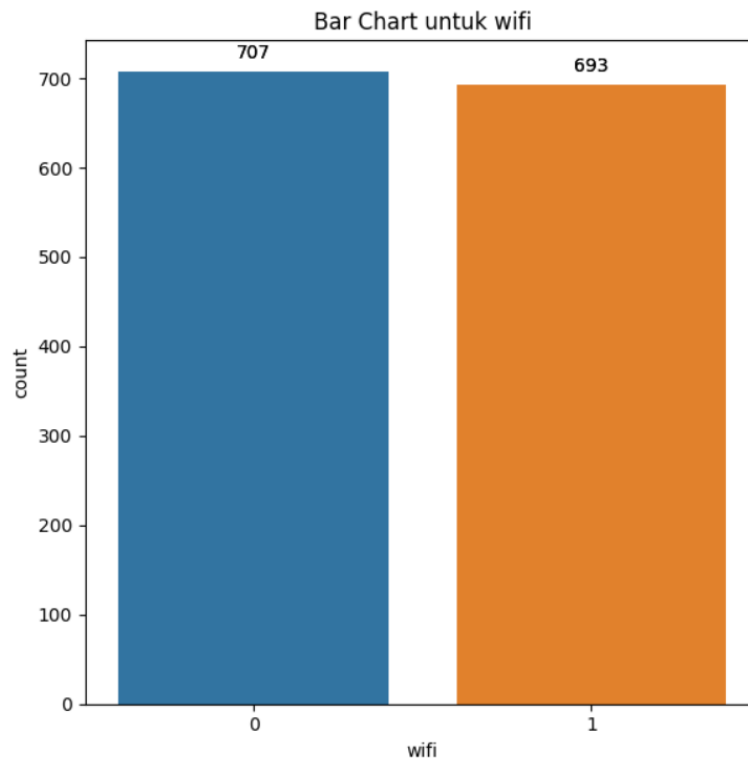
19. *touch_screen*: Memiliki layar sentuh atau tidak (non numerik)



Gambar 5.19 Perbandingan Jumlah Nilai Kolom *touch_screen*

Kolom *touch_screen* tergolong ke dalam kolom non numerik dengan nilai 1 = *true* dan 0 = *false*. Perbandingan frekuensi kedua nilai dapat dilihat pada **Gambar 5.19** di atas. Frekuensi nilai 0 sedikit lebih banyak dibandingkan frekuensi nilai 1. Artinya, ponsel yang tidak memiliki layar sentuh pada data latih jumlahnya sedikit lebih banyak dibanding dengan ponsel yang memiliki layar sentuh.

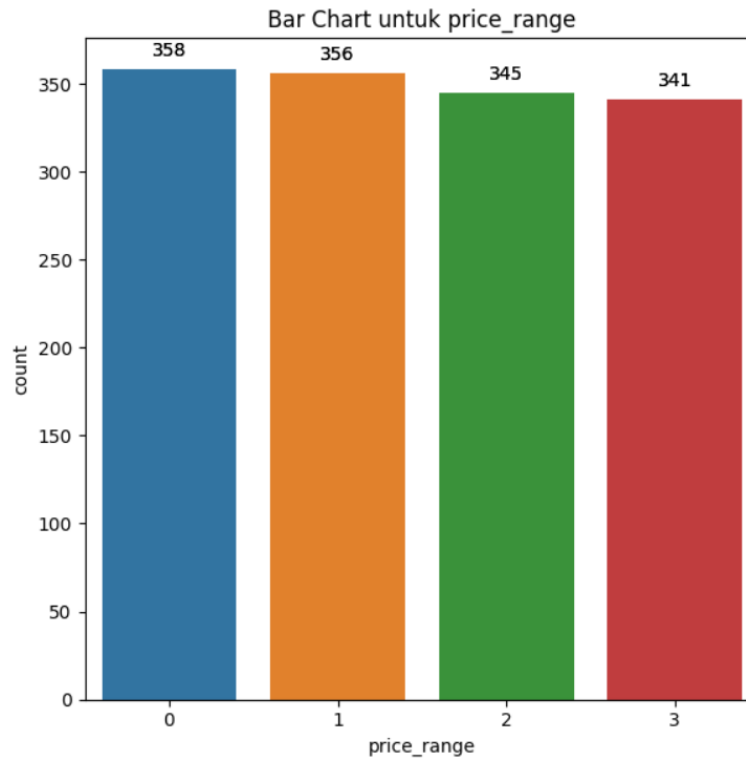
20. *wifi*: Memiliki wifi atau tidak (non numerik)



Gambar 5.20 Perbandingan Jumlah Nilai Kolom *wifi*

Kolom *wifi* tergolong ke dalam kolom non numerik dengan nilai 1 = *true* dan 0 = *false*. Perbandingan frekuensi kedua nilai dapat dilihat pada **Gambar 5.20** di atas. Frekuensi nilai 0 sedikit lebih banyak dibandingkan frekuensi nilai 1. Artinya, ponsel yang tidak memiliki *wifi* pada data latih jumlahnya sedikit lebih banyak dibanding dengan ponsel yang memiliki *wifi*.

21. *price_range* (**target**): Rentang harga dengan nilai 0 (biaya rendah), 1 (biaya sedang), 2 (biaya tinggi) atau 3 (biaya sangat tinggi) (non numerik)



Gambar 5.21 Perbandingan Jumlah Nilai Kolom *price_range*

Kolom *price_range* sebagai kolom target juga tergolong ke dalam kolom non numerik dengan kategori nilai 0 berarti ponsel dengan biaya rendah, 1 berarti ponsel dengan biaya sedang, 2 berarti ponsel dengan biaya tinggi, atau 3 berarti ponsel dengan biaya sangat tinggi. Perbandingan frekuensi nilai-nilai tersebut dapat dilihat pada **Gambar 5.21** di atas. Terlihat frekuensi untuk nilai-nilai tersebut menurun secara berurut dari kategori nilai 0, 1, 2, dan 3. Artinya, ponsel dengan harga rendah masih lebih banyak dibanding ponsel harga lainnya, dan ponsel dengan harga sangat tinggi memiliki jumlah yang paling sedikit, meskipun sebaran keempat kategori ini tidak terlalu jauh.

6. Korelasi dengan Kolom Target

Seperti yang telah disebutkan sebelumnya, pada data latih yang digunakan kolom targetnya adalah kolom terakhir yakni *price_range*, yang menunjukkan kategori harga ponsel pada data latih. Pada bagian ini, akan dianalisis bagaimana korelasi antara atribut-atribut non target terhadap atribut target *price_range* tersebut dengan menggunakan bantuan *library* pandas.

```
# target kolom price_range
target_column = 'price_range'
correlation_matrix = df.corr()
correlation_with_target = correlation_matrix[target_column]

# display
print("Korelasi setiap kolom ke kolom target:")
print(correlation_with_target)
```

Dan diperoleh nilai korelasi masing-masing atribut sebagai berikut.

```
Korelasi setiap kolom ke kolom target:
battery_power      0.184801
blue               0.041947
clock_speed        0.014031
dual_sim           -0.010756
fc                 -0.003842
four_g             0.000551
int_memory         0.026176
m_dep              0.001205
mobile_wt          -0.074769
n_cores            -0.000582
pc                 -0.005214
px_height          0.158833
px_width           0.178713
ram                0.918319
sc_h               0.012149
sc_w               0.019912
talk_time          0.011113
three_g            0.027098
touch_screen       -0.029842
wifi               0.034329
price_range        1.000000
Name: price_range, dtype: float64
```

Gambar 6.1 Hasil Korelasi Setiap Kolom terhadap Kolom Target

Nilai-nilai korelasi tersebut dapat menunjukkan bagaimana keterkaitan sebuah kolom dengan kolom target:

1. Jika nilainya mendekati 1, maka nilai pada kolom tersebut akan berbanding lurus dengan nilai pada kolom target. Artinya, jika sebuah nilai pada kolom yang berkorelasi tersebut bernilai sangat tinggi, maka nilai pada kolom target juga akan ikut naik. Begitupun sebaliknya jika nilai pada sebuah kolom yang berkorelasi tersebut turun, maka nilai pada kolom target juga akan ikut turun.
2. Jika nilainya mendekati -1, maka nilai pada kolom tersebut akan berbanding terbalik dengan nilai pada kolom target. Artinya, jika sebuah nilai pada kolom yang berkorelasi tersebut bernilai sangat tinggi, maka nilai pada kolom target justru akan menurun. Begitupun sebaliknya jika nilai pada sebuah kolom yang berkorelasi tersebut turun, maka nilai pada kolom target justru akan menaik.
3. Jika nilainya mendekati 0, maka kolom tersebut cenderung tidak berkorelasi/berpengaruh apa-apa terhadap nilai kolom target.

Jika dilihat pada hasil analisis yang dilakukan, dapat diamati:

1. Kolom-kolom yang berkorelasi berbanding lurus dengan kolom target (mendekati 1) antara lain kolom *ram* (0.9) dan kolom target itu sendiri (sudah tentu).
2. Kolom-kolom yang berkorelasi berbanding terbalik dengan kolom target (mendekati -1) tidak ada
3. Semua kolom selain *ram* dan *price_range* memiliki nilai korelasi mendekati 0 sehingga cenderung memiliki korelasi nilai yang tidak berpengaruh terhadap nilai pada kolom target.

Jadi, dapat disimpulkan bahwa nilai yang paling berpengaruh terhadap tingkat harga sebuah ponsel adalah RAM dari ponsel tersebut. Semakin besar RAM, semakin mahal harga sebuah ponsel.

Referensi

1. Handling Missing Values in Pandas Dataframe | GeeksforGeeks:
<https://youtu.be/uDr67HBIPz8?si=-yyozSBiLPLi6dDK>
2. Find duplicate rows in a Dataframe based on all or selected columns:
<https://www.geeksforgeeks.org/find-duplicate-rows-in-a-dataframe-based-on-all-or-selected-columns/>
3. Python Pandas Tutorial 19 | How to Identify and Drop Duplicate Values | Removing duplicate values: <https://youtu.be/ix9iGffOA5U?si=aRwBfewE9Cn3-WCY>
4. Exploratory Data Analysis using Pandas Profiling in Jupyter Notebook:
<https://youtu.be/WmDgLGQeRbQ?si=ggorpsy9kPbtWP4H>
5. How to Detect and Remove Outliers in the Data | Python:
<https://youtu.be/Cw2lvmWRcXs?si=b324hqbZ1OrGTrjX>
6. Kenali Analisis Statistik dalam Ukuran Penyebaran Data:
<https://dqqlab.id/kenali-analisis-statistik-dalam-ukuran-penyebaran-data>
7. Transformations to reduce skewness and kurtosis: [manvendra7/Skewness-and-kurtosis: Transformations to reduce skewness and kurtosis \(github.com\)](https://github.com/manvendra7/Skewness-and-kurtosis)
8. Exploratory Data Analysis (EDA) dengan Python:
<https://arinannp.medium.com/exploratory-data-analysis-eda-dengan-python-5c99ec8d9934>