Scrapping harga GPU RTX 3070 di Tokopedia menggunakan python

Muhammad Arya Danuarta

09011282025035

Proses ini menggunakan file html yang diunduh secara langsung dari website target yang ingin diambil datanya. Hal ini dilakukan karena response web pada situs aslinya menggunakan lazy loading sehingga membatasi pengambilan data karena website harus dinavigasi keseluruhan agar data yang diinginkan termuat. Menggunakan bantuan beberapa library yang bisa dilihat dibawah ini

In [1]:
```
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
list_item = []
list_harga = []
list_total = []
```

In [3]:
```
with open('Jual rtx 3070 _ Tokopedia.html', 'r') as html_file:
    html = html_file.read()

soup = BeautifulSoup(html, 'lxml')
items = soup.find_all('div', class_='css-12sieg3')
```

In [4]:
```
for item in items:
    nama_item = item.find('div', class_='css-1b6t4dn').text.replace(',', '')
    harga_item = item.find('div', class_='css-1ksb19c').text.replace('Rp', '')

    list_total.append(nama_item)
    list_total.append(harga_item.replace('.', ''))
```

In [5]:
```
f = open('hasil.csv', 'w')
f.write("Nama Barang,Harga\n")

for i in range(len(list_total)):
    if i % 2 == 0:
        f.write((list_total[i] + ','))
        f.write(list_total[i+1])
    else:
        f.write("\n")

f.close()
```

In [6]:
```
df = pd.read_csv('hasil.csv')
```

In [7]:
```
df.head(5)
```

Out[7]:

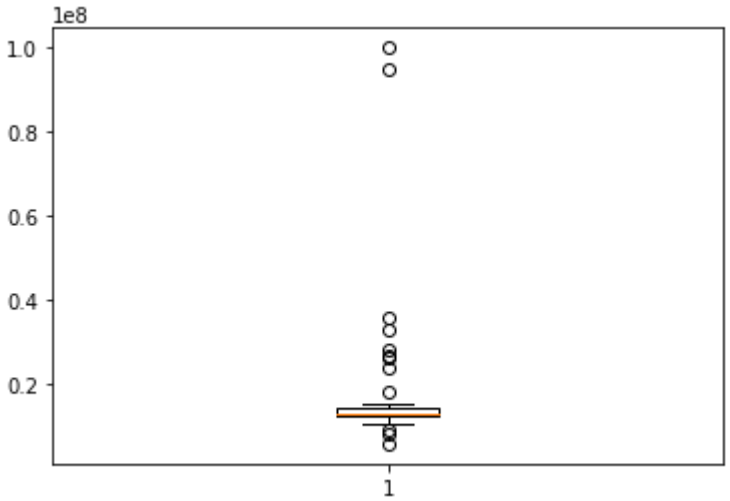
	Nama Barang	Harga
0	Gigabyte Nvidia GeForce RTX 3070 GAMING OC 8GB...	12831000
1	Vga Zotac Geforce RTX 3050 Twin Edge 8GB GDDR6...	5880000
2	Palit Nvidia GeForce RTX 3070 8GB GamingPro - ...	11930000
3	ASUS GeForce RTX 3070 Noctua Edition OC 8GB GD...	13190000
4	Gainward GeForce RTX™ 3070 Ti Phoenix	13899000

In [8]:
```
df.dtypes
```

Out[8]:
```
Nama Barang    object
Harga           int64
dtype: object
```

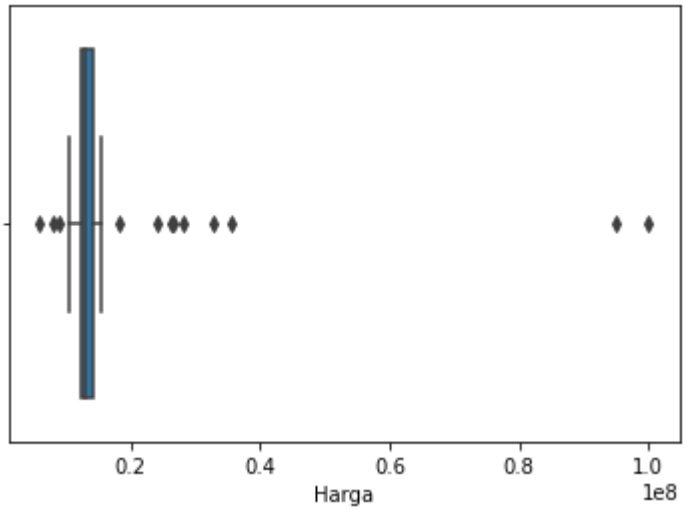
In [9]:
```
plt.boxplot(df['Harga'])
```

Out[9]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x116870a8a90>,
  <matplotlib.lines.Line2D at 0x116870a8df0>],
 'caps': [<matplotlib.lines.Line2D at 0x116870c1190>,
  <matplotlib.lines.Line2D at 0x116870c14f0>],
 'boxes': [<matplotlib.lines.Line2D at 0x116870a8730>],
 'medians': [<matplotlib.lines.Line2D at 0x116870c1850>],
 'fliers': [<matplotlib.lines.Line2D at 0x116870c1bb0>],
 'means': []}
```



In [10]:
```
sns.boxplot(x=df['Harga'])
```

Out[10]:
```
<AxesSubplot:xlabel='Harga'>
```



Kemungkinan besar terdapat outlier yang perlu dieleminasi terlebih dahulu pada dataset, proses dibawah merupakan eliminasi outlier dengan IQR

In [11]:
```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

```
Harga    1728000.0
dtype: float64
```

In [12]:
```
import warnings
warnings.simplefilter(action="ignore", category=FutureWarning)
compare = [(df < (Q1 - 1.5 * IQR))|(df > (Q3 + 1.5 * IQR))]
print(compare)
```

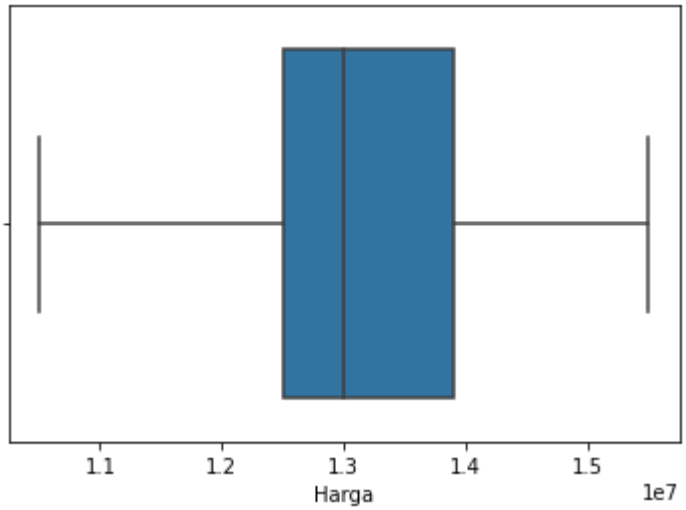
```
[    Harga  Nama Barang
0   False        False
1    True        False
2   False        False
3   False        False
4   False        False
..    ...          ...
75   True        False
76  False        False
77   True        False
78  False        False
79  False        False

[80 rows x 2 columns]]
```

In [13]:
```
df_out = df[~((df < (Q1 - 1.5 * IQR)) |(df > (Q3 + 1.5 * IQR))).any(axis=1)]
df_out.shape
```

Out[13]: (68, 2)

In [14]:
```
sns.boxplot(x=df_out['Harga'])
```

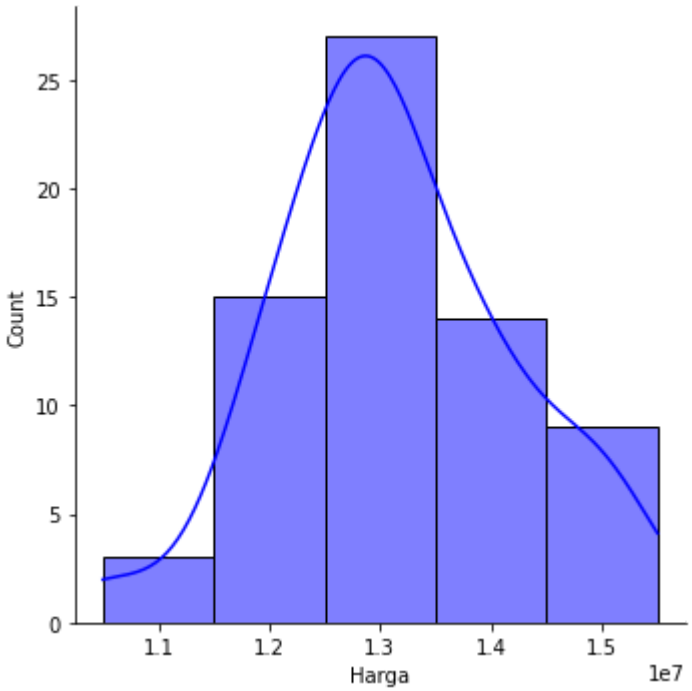
Out[14]: <AxesSubplot:xlabel='Harga'>



Sekarang dataset lebih tertata tanpa outlier

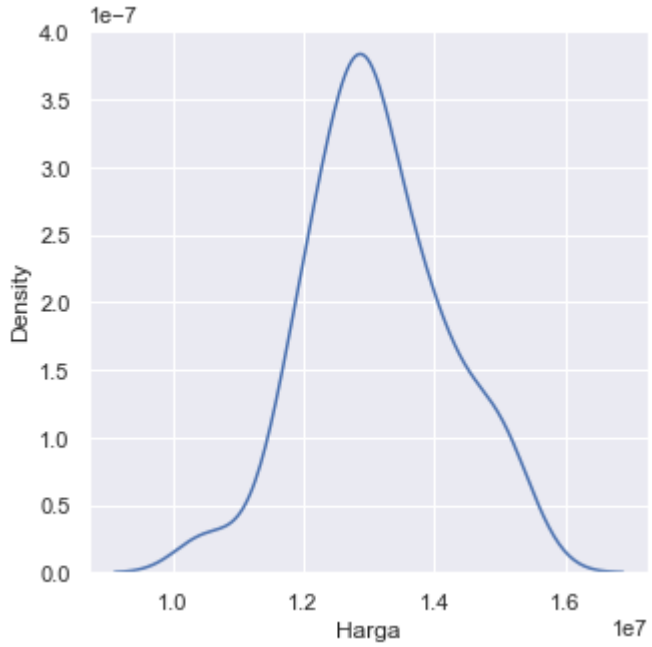
In [15]:
```
sns.displot(df_out['Harga'],kde=True, color='blue', bins=5)
```

Out[15]: <seaborn.axisgrid.FacetGrid at 0x11687b208b0>



In [16]:
```
sns.set(rc={'figure.figsize':(5,5)})
sns.kdeplot(df_out['Harga'],shade=False)
```

Out[16]: <AxesSubplot:xlabel='Harga', ylabel='Density'>



In [125…

```
df_out.mean()
```

Out[125…

```
Harga    1.312107e+07
dtype: float64
```

Rata-rata harga barang terdapat di 1.312107e+07 atau = Rp. 13.121.070

In [ ]: