LAPORAN UJIAN TENGAH SEMESTER

PRAKTIKUM KOMPUTASI STATISTIKA II

KELAS A

Yogyakarta, 20 April 2023

Anggota Kelompok : 1. Allisya Maharani Adinda Wibowo

2. Bryan Florentino Leo

3. Natasya Fatimah Salim

Dosen Pengampu : Prof. Dr.rer.nat. Dedi Rosadi, S.Si., M.Sc.

Asisten Praktikum : 1. Iqbal Hanif Anggita Adi (19542)

2. R.Bg. Rifaat Puthut Guritno (20033)

LABORATORIUM KOMPUTASI

MATEMATIKA DAN STATISTIKA

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS GADJAH MADA

2023

BAB I
PERMASALAHAN

Permasalahan untuk Kelompok

1. Lakukan *web scraping* pada *website* apapun. Jelaskan tahapan berpikir dalam membuat *code web scraping* tersebut. Kemudian, jelaskan bagian-bagian dari *code* yang telah dibuat.

2. Dapatkan data menggunakan suatu API. Jelaskan cara penggunaan API tersebut beserta *code*-nya.

3. Lakukan *data preprocessing* dan visualisasikan data hasil soal nomor 1 dan 2 serta jelaskan *code*-nya. Kemudian, buat interpretasi berdasarkan visualisasi data tersebut.

Permasalahan untuk Individu

Jelaskan kontribusi masing-masing anggota kelompok terhadap tugas yang dikerjakan dan berilah penilaian (0 – 100) kepada masing-masing anggota kelompok (nilai tidak boleh ada yang sama). Kumpulkan melalui *form*: https://forms.office.com/r/UX9bYJhT50.

BAB II
PEMBAHASAN

A. PEMBAHASAN MASALAH NOMOR 1

Diketahui:
Web scraping.

Ditanya:
- *Web scraping* pada *website* apapun.
- Penjelasan tahapan berpikir dalam membuat *code web scraping*.
- Penjelasan bagian-bagian dari *code*.

Jawab:
Dari *link* https://www.webometrics.info/en/asia/indonesia%20, dilakukan *web scraping* dengan tahapan berpikir sebagai berikut.

1. Menentukan *website* yang akan di-*scraping* (dengan memperhatikan ketentuan apakah *website* tersebut boleh di-*scraping* atau tidak, informasi apa yang akan diambil, dan apakah informasi tersebut akan berguna atau tidak).
2. Menganalisis *website* tersebut untuk mengetahui struktur HTML dan URL-nya.
3. Menentukan *module* yang akan digunakan dengan menyesuaikan *website* yang akan di-*scraping*. Contoh *module* yang umum digunakan adalah `BeautifulSoup`, `Selenium`, `Scrapy`, dan `Requests`.
4. Membuat *code scraping* yang bisa membaca struktur HTML dari website tersebut dan mengambil data yang diinginkan. Kemudian, melakukan iterasi atau mengulang proses *scraping* pada seluruh halaman di *website* tersebut hingga diperoleh seluruh data.
5. Memeriksa kembali kesesuaian hasil *scraping*.

Code beserta penjelasannya terlampir sebagai berikut.

```
#Import library yang diperlukan
import requests
from bs4 import BeautifulSoup
import pandas as pd

# Menginisialisasi list kosong untuk menyimpan hasil scraping
universities = []

# Melakukan iterasi pada seluruh halaman di
https://www.webometrics.info/en/asia/indonesia%20
for page in range(0, 35):
    url = f"https://www.webometrics.info/en/asia/indonesia%20?page={page}"
    response = requests.get(url)

    # Memeriksa apakah status response adalah 200 (dapat discraping)
    if response.status_code == 200:
        soup = BeautifulSoup(response.content, "html.parser")
```

```
        # Menemukan tabel universitas pada halaman dan mengambil datanya
        table = soup.find("table", {"class": "sticky-enabled"})
        rows = table.find_all("tr")[1:]

        # Menyimpan data universitas pada list
        for row in rows:
            cells = row.find_all("td")
            rank = cells[0].text.strip()
            worldrank = cells[1].text.strip()
            name = cells[2].text.strip()
            impact = cells[4].text.strip()
            openness = cells[5].text.strip()
            excellence = cells[6].text.strip()

            universities.append({
                "Rank": rank,
                "World Rank": worldrank,
                "University": name,
                "Impact": impact,
                "Openness": openness,
                "Excellence": excellence
            })
    else:
        print(f"Failed to retrieve page {page}")

# Membuat data frame dari list universities
Univ = pd.DataFrame(universities)
```

B. PEMBAHASAN MASALAH NOMOR 2

Diketahui:
Application Programming Interface (API).

Ditanya:
- Mendapatkan data dari API apapun.
- Penjelasan cara penggunaan API beserta *code*.

Jawab:
Dari *link* https://www.sephora.co.id/categories/makeup/lips?page=7, ingin didapatkan data menggunakan suatu API dengan *code* beserta penjelasannya sebagai berikut.

```
#import library yang diperlukan
import requests
import pandas as pd

#membuat array kosong untuk menyimpan list product
product = []

#melakukan perulangan untuk mengakses data produk dari halaman 1 hingga halaman 7
for i in range (1,8):
	page = i
	url =
'https://www.sephora.co.id/api/v2.3/products?filter[category]=makeup%2Flips&page[n
umber]={}&page[size]=36&sort=sales&include=variants,brand,featured_ad'.format(page
)
	response = requests.get(url).json()
	data = response['data']
	#menyiapkan array kosong untuk menyimpan atribut yang ada di dalam segmen
data
	at = []
	#melakukan perulangan untuk mengakses atribut yang ada di dalam segmen data
	for p in data:
		#menginputkan setiap atribut yang ada di dalam segmen data
		at.append(p['attributes'])
		#melakukan perulangan untuk mengakses keterangan produk yang ada di
dalam atribut
		for t in at:
		#mengambil data-data keterangan produk yang diperlukan dari website
			nama = t['name']
			harga = t['price']
			rating = t['rating']
			review = t['reviews-count']
			varian = t['variants-count']
			isi = t['heading']
			if t['sold-out'] == True:
				t['sold-out'] = 'Sold-Out'
			elif t['sold-out'] == False:
				t['sold-out'] = 'Available'
```

```
                    keterangan = t['sold-out']
                    if t['under-sale'] == True:
                        t['under-sale'] = 'Sale'
                    elif t['under-sale'] == False:
                        t['under-sale'] = 'Normal'
                    price_info = t['under-sale']

        #menginputkan setiap keterangan produk ke dalam array 'product'
        product.append((nama, harga, rating, review, varian, isi, price_info))
        #membuat dataframe untuk setiap produk
        prod = pd.DataFrame(product, columns=['nama', 'harga', 'rating', 'review',
'banyak varian', 'isi', 'price info'])
```

C. PEMBAHASAN MASALAH NOMOR 3

Diketahui:
Data preprocessing dan visualisasi data.

Ditanya:
- *Data preprocessing* dan visualisasi data terhadap *dataset* yang diperoleh dari nomor 1 dan 2.
- Penjelasan *code*.
- Interpretasi.

Jawab:
a. *Dataset* Nomor 1
Setelah diperoleh *dataframe* dari nomor 1, dilakukan *data preprocessing* untuk menyiapkan data agar bisa dianalisis.

Preprocessing pertama yang dilakukan adalah menampilkan informasi dari setiap variabel untuk memeriksa adanya perbedaan jumlah *tuple* dan tipe data. Sintaks yang dijalankan adalah sebagai berikut.

```
#Menampilkan info data
univ.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3462 entries, 0 to 3461
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Rank        3462 non-null   object
 1   World Rank  3462 non-null   object
 2   University  3462 non-null   object
 3   Impact      3462 non-null   object
 4   Openness    3462 non-null   object
 5   Excellence  3462 non-null   object
dtypes: object(6)
memory usage: 162.4+ KB
```

Berdasarkan keluaran di atas, semua variabel memuat 3.462 *tuple* yang tidak bernilai *null* dan bertipe data *object* (*string*). Agar dapat divisualisasikan, tipe data dari beberapa variabel diubah menjadi *integer*.

```
#Memperbaiki tipe data
univ.Rank = univ.Rank.astype(int)
univ["World Rank"] = univ["World Rank"].astype(int)
univ.Impact = univ.Impact.astype(int)
univ.Openness = univ.Openness.astype(int)
univ.Excellence = univ.Excellence.astype(int)
```

Setelah itu, dipastikan kembali apakah terdapat data dengan *missing value* dan berduplikat.

```
#Mengecek adanya missing value
missing_data = pd.DataFrame({'total_missing': univ.isnull().sum(),
                             'perc_missing': (univ.isnull().sum()/12006)*100})
missing_data
```

	total_missing	perc_missing
Rank	0	0.0
World Rank	0	0.0
University	0	0.0
Impact	0	0.0
Openness	0	0.0
Excellence	0	0.0

```
#Memeriksa adanya data duplikat
univ[univ.duplicated()]
```

Rank	World Rank	University	Impact	Openness	Excellence

Disimpulkan bahwa tidak terdapat data yang *missing* atau berduplikat satu sama lain. Selanjutnya, diamati deskripsi numerik dari data sebagai berikut.

```
#Mengetahui deskripsi numerik data
univ.describe()
```

	Rank	World Rank	Impact	Openness	Excellence
count	3462.000000	3462.000000	3462.000000	3462.000000	3462.000000
mean	1731.944541	23874.708550	23943.134893	6236.235991	7144.655690
std	1000.293546	8066.692124	8252.584750	889.228086	501.963973
min	1.000000	583.000000	193.000000	686.000000	1126.000000
25%	866.250000	19197.250000	19945.000000	6553.000000	7212.000000
50%	1731.500000	26801.000000	26802.000000	6553.000000	7212.000000
75%	2596.750000	30444.000000	30430.000000	6553.000000	7212.000000
max	3481.000000	32230.000000	32229.000000	6553.000000	7212.000000

Setelah melakukan *preprocessing*, data yang dimiliki sudah siap untuk divisualisasikan.

Dari *data frame* tersebut, akan dilakukan beberapa visualisasi data sebagai berikut.

1. *Distribution plot* untuk mengetahui distribusi variabel "World Rank", “Impact”, “Openness”, dan “Excellence”.
2. *Boxplot* untuk melihat distribusi variabel "World Rank”, “Impact”, “Openness”, dan “Excellence” secara lebih mendetail serta mengamati adanya *outlier* pada setiap variabel tersebut.

3. *Heatmap* korelasi untuk mengamati nilai kekuatan hubungan antarvariabel dari variabel-variabel "World Rank", "Impact", "Openness", dan "Excellence".
4. *Bar chart* untuk mengetahui jumlah data dari setiap kategori pada variabel baru "World Rank Category".

Visualisasi data melibatkan dua buah *module*, yaitu *submodule* pyplot dari *module* matplotlib dan *module* seaborn.

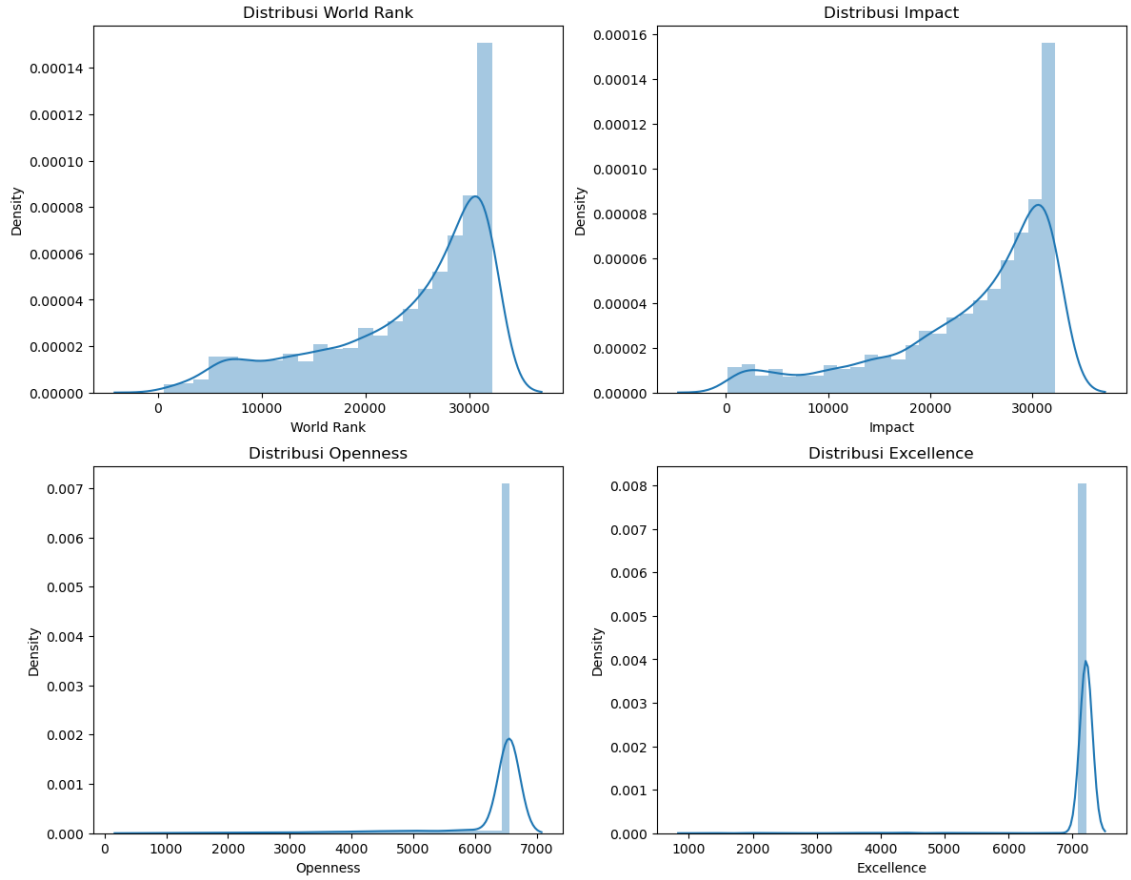
```
#Mengimpor module yang diperlukan
import matplotlib.pyplot as plt
import seaborn as sns
```

Distribution plot

Sintaks:

```
fig, axs = plt.subplots(2, 2, figsize=(14, 11))

#Distribution plot variabel 'World Rank'
sns.distplot(univ["World Rank"], ax = axs[0][0])
axs[0][0].set_title('Distribusi World Rank')

#Distribution plot variabel 'Impact'
sns.distplot(univ["Impact"], ax = axs[0][1])
axs[0][1].set_title('Distribusi Impact')

#Distribution plot variabel 'Openness'
sns.distplot(univ["Openness"], ax = axs[1][0])
axs[1][0].set_title('Distribusi Openness')

#Distribution plot variabel 'Excellence'
sns.distplot(univ["Excellence"], ax = axs[1][1])
axs[1][1].set_title('Distribusi Excellence')

plt.show()
```

Keluaran:



Interpretasi:

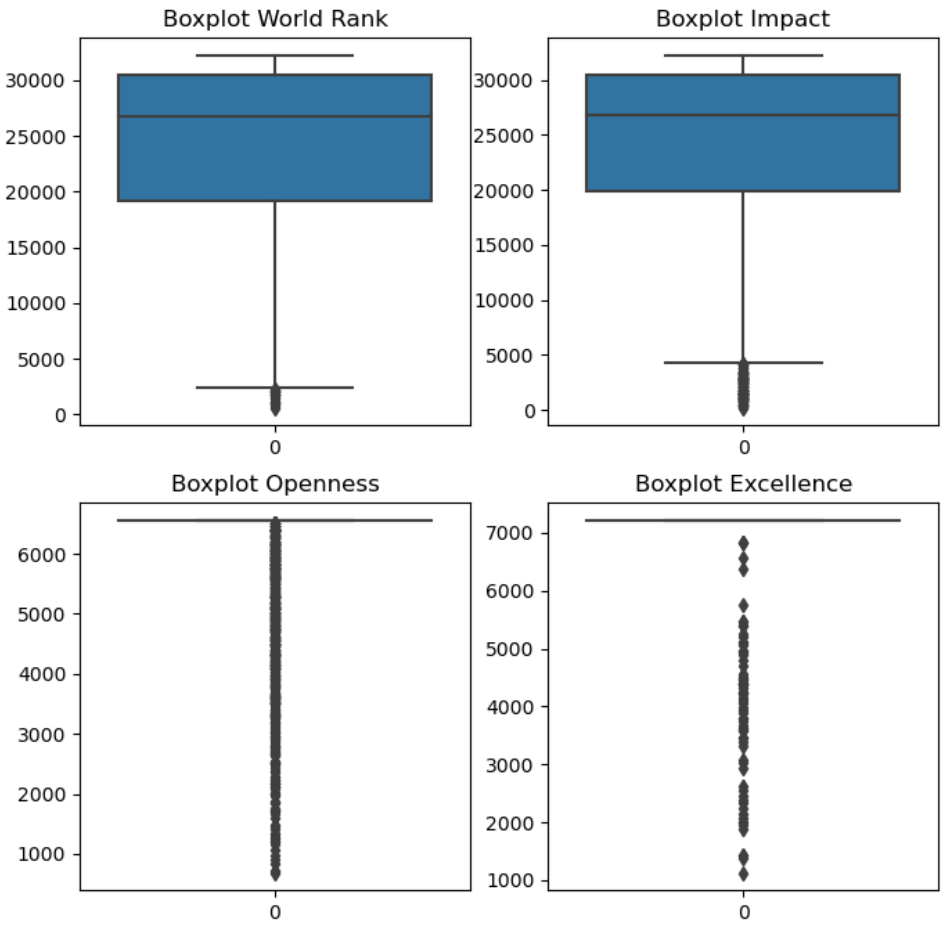
- Distribusi data variabel “World Rank” menceng ke kiri (*negative skewed*) dengan puncak pada nilai sekitar 30.000. Diketahui rata-rata dari “World Rank” senilai 23.874,708550, deviasi standar senilai 8.066,692124, nilai minimum 583, dan nilai maksimum 32.230. Artinya, kebanyakan perguruan tinggi di Indonesia berada pada peringkat 30.000-an.
- Distribusi data variabel “Impact” menceng ke kiri (*negative skewed*) dengan puncak pada nilai sekitar 30.000. Diketahui rata-rata dari “Impact” senilai 23.943,134893, deviasi standar senilai 8.252,584750, nilai minimum 193, dan nilai maksimum 32.229. Artinya, dampak eksternal kebanyakan perguruan tinggi di Indonesia berada pada peringkat 30.000-an.
- Distribusi data variabel “Openness” menceng ke kiri (*negative skewed*) dengan puncak pada nilai sekitar 6.500. Diketahui rata-rata dari “Openness” senilai 6.236,235991, deviasi standar senilai 889,228086, nilai minimum 686, dan nilai maksimum 6.553. Artinya, keterbukaan kebanyakan perguruan tinggi di Indonesia berada pada peringkat 6.500-an.
- Distribusi data variabel “Excellence” menceng ke kiri (*negative skewed*) dengan puncak pada nilai sekitar 7.000. Diketahui rata-rata dari “Excellence” senilai 7.144,655690, deviasi standar senilai 501,963973, nilai minimum 1.126, dan nilai maksimum 7.212. Artinya, kecemerlangan kebanyakan perguruan tinggi di Indonesia berada pada peringkat 7.000-an.

Boxplot

Sintaks:

```
fig, axs = plt.subplots(2, 2, figsize=(8, 8))

#Boxplot variabel 'World Rank'
sns.boxplot(data = univ["World Rank"], ax = axs[0][0])
axs[0][0].set_title('Boxplot World Rank')

#Boxplot variabel 'Impact'
sns.boxplot(data = univ["Impact"], ax = axs[0][1])
axs[0][1].set_title('Boxplot Impact')

#Boxplot variabel 'Openness'
sns.boxplot(data = univ["Openness"], ax = axs[1][0])
axs[1][0].set_title('Boxplot Openness')

#Boxplot variabel 'Excellence'
sns.boxplot(data = univ["Excellence"], ax = axs[1][1])
axs[1][1].set_title('Boxplot Excellence')
```

Keluaran:



Interpretasi:

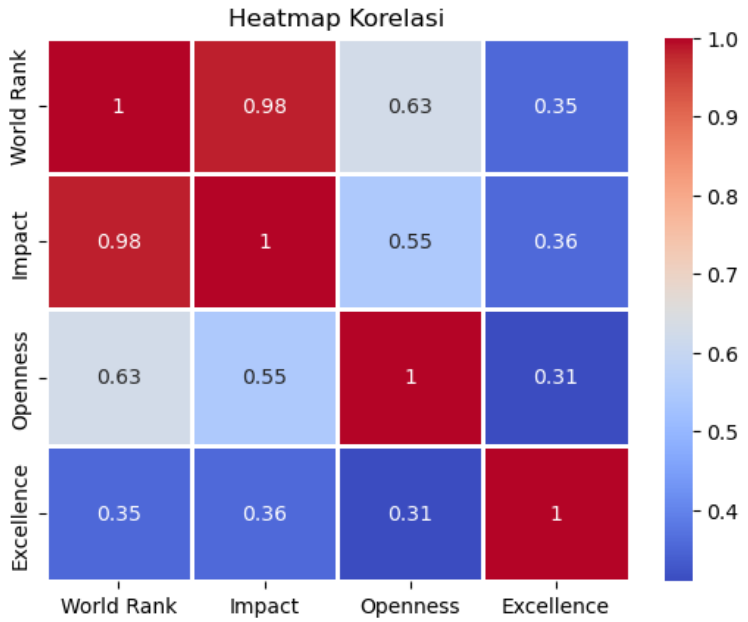
- Distribusi data variabel “World Rank” menjurai ke bawah (*negative skewed*) dengan memuat sedikit pencilan bawah. Diketahui nilai kuartil bawah, median, dan kuartil atas dari variabel “World Rank” berturut-turut sebesar 866,25, 1.731,5, dan 2.596,75. Kotak yang berkorespondensi dengan nilai-nilai kuartil ini tidak simetris, sehingga dapat ditafsirkan bahwa distribusi data tidak simetris.
- Distribusi data variabel “Impact” menjurai ke bawah (*negative skewed*) dengan memuat sedikit pencilan bawah. Diketahui nilai kuartil bawah, median, dan kuartil atas dari variabel “Impact” berturut-turut sebesar 19.945, 26.802, dan 30.430. Kotak yang berkorespondensi dengan nilai-nilai kuartil ini tidak simetris, sehingga dapat ditafsirkan bahwa distribusi data tidak simetris.
- Distribusi data variabel “Openness” menjurai ke bawah (*negative skewed*) dengan memuat hampir semua data sebagai pencilan bawah. Diketahui nilai kuartil bawah, median, dan kuartil atas dari variabel “Openness” ketiganya sebesar 6.553. Karena sama dengan nilai maksimum, maka garis-garis yang membentuk kotak *boxplot* saling berimpitan, sehingga kotak tidak tampak.
- Distribusi data variabel “Excellence” menjurai ke bawah (*negative skewed*) dengan memuat hampir semua data sebagai pencilan bawah. Diketahui nilai kuartil bawah, median, dan kuartil atas dari variabel “Impact” ketiganya sebesar 7.212. Karena sama dengan nilai maksimum, maka garis-garis yang membentuk kotak *boxplot* saling berimpitan, sehingga kotak tidak tampak.

***Heatmap* Korelasi**

Sintaks:

```
univ1 = univ[["World Rank", "Impact", "Openness", "Excellence"]]
sns.heatmap(univ1.corr(), cmap = 'coolwarm', annot = True, linecolor = 'white', linewidths = 1).set_title("Heatmap Korelasi")
```

Keluaran:



Interpretasi:

Korelasi terbesar terdapat antara variabel “World Rank” dengan “Impact”, yaitu sebesar 0,98. Artinya, terdapat hubungan positif yang sangat kuat antara kedua variabel ini. Meningkatnya nilai salah satu variabel berdampak pada meningkatnya nilai variabel yang lain dengan proporsi yang hampir sama.

Korelasi terkecil terdapat antara variabel “Excellence” dengan “Openness”, yaitu sebesar 0,31. Artinya, terdapat hubungan positif yang lemah antara kedua variabel ini. Meningkatnya nilai salah satu variabel berdampak pada meningkatnya nilai variabel yang lain.

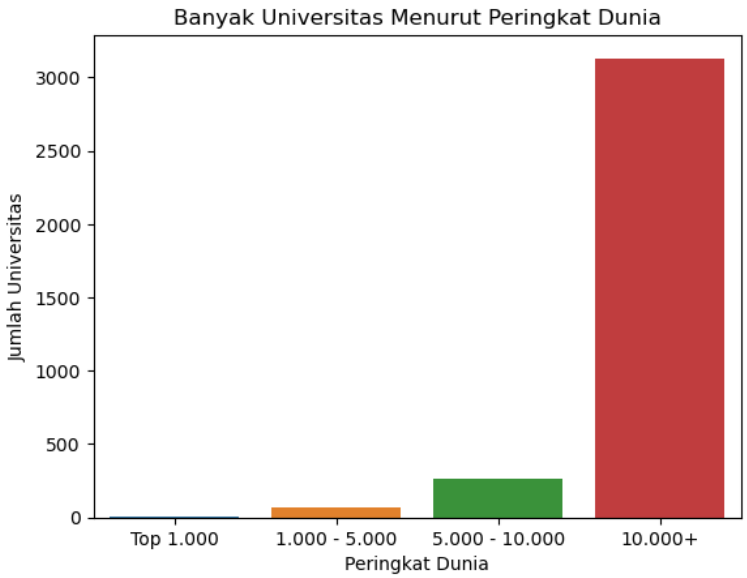
Bar chart

Sintaks:

```
#Membuat kolom baru, World Rank Category
WRC = []
for i in range(len(univ)):
    if univ["World Rank"][i] <= 1000:
        WRC.append("Top 1.000")
    elif univ["World Rank"][i] > 1000 and univ["World Rank"][i] <= 5000:
        WRC.append("1.000 - 5.000")
    elif univ["World Rank"][i] > 5000 and univ["World Rank"][i] <= 10000:
        WRC.append("5.000 - 10.000")
    else:
        WRC.append("10.000+")

univ["World Rank Category"] = WRC
```

```
#Bar chart dari variabel 'World Rank Category'
sns.countplot(data = univ, x = 'World Rank Category')
plt.title('Banyak Universitas Menurut Peringkat Dunia')
plt.xlabel('Peringkat Dunia')
plt.ylabel('Jumlah Universitas')
plt.show()
```

Keluaran:



Interpretasi:

Dibentuk variabel baru yang membagi peringkat dunia menjadi empat kategori, yaitu Top 1.000, 1.000 - 5.000, 5.000 - 10.000, dan 10.000+. Dapat disimpulkan bahwa mayoritas (lebih dari tiga ribu) perguruan tinggi di Indonesia menempati peringkat dunia di atas 10.000.

b. *Dataset* Nomor 2

Setelah diperoleh *dataframe* dari nomor 2, dilakukan *data preprocessing* untuk menyiapkan data agar bisa dianalisis.

Preprocessing pertama yang dilakukan adalah menentukan apakah tipe data untuk setiap variabel sudah sesuai dan apakah ada *missing value* pada data sebagai berikut.

```
#Menentukan apakah tipe data sudah sesuai atau belum
prod.dtypes
```

```
nama             object
harga             int64
rating          float64
review            int64
banyak varian     int64
isi              object
price info       object
dtype: object
```

```
[17] #Mengecek apakah terdapat missing value pada data
     print(prod.isna().sum())
```

```
nama             0
harga            0
rating           0
review           0
banyak varian    0
isi             44
price info       0
dtype: int64
```

Dari *output* di atas, dapat diketahui bahwa tipe data untuk variabel "isi" belum sesuai dan terdapat 44 *missing value* pada variabel "isi". Dikarenakan pada analisis berikutnya variabel "isi" tidak digunakan, variabel "isi" dapat dihapus sebagai berikut.

```
[ ] #Dikarenakan kita tidak akan menggunakan variabel 'isi' dan variabel ini mengandung banyak missing value, maka variabel ini akan dihapus
    prod = prod.drop('isi', axis = 1)
    prod
```

	nama	harga	rating	review	banyak varian	price info
0	Soft Pinch Tinted Lip Oil	145000	4.7	846	8	Normal
1	Lip Soufflé Matte Lip Cream	145000	4.4	1979	15	Normal
2	Stay Vulnerable Glossy Lip Balm	145000	4.7	1140	5	Normal
3	Poutsicle Hydrating Lip Stain	182000	4.4	791	4	Normal
4	Benetint Cheek & Lip Tint	130000	4.2	1271	1	Normal
...	...	...	...	...	...	...
247	L'Absolu Lacquer Liquid Lipstick X Chiara Ferr...	199955	4.0	4	3	Normal
248	Love Moi Pink Lip Balm	120000	4.3	11	1	Normal
249	Be Mine Beauty Makeup Set	239000	4.6	14	1	Normal
250	Watermelon Burst Hydrating Lip Oil	101000	4.0	79	1	Normal
251	Lip Honeys Colorful Gloss Balm	78000	4.3	317	2	Normal

252 rows × 6 columns

Kemudian, akan dicek apakah terdapat duplikat data.

```
[27] #Memeriksa adanya data duplikat
     prod[prod.duplicated()]
```

	nama	harga	rating	review	banyak varian	price info
180	Allure Shine Lustrous Lip Plumper	225000	3.8	12	3	Normal
181	Squalane + Rose Vegan Lip Balm	193000	3.9	284	1	Normal
182	Aroma Lipstick	125000	4.0	2	7	Normal

```
#Menghapus data duplikat
prod.drop_duplicates(inplace=True)
```

Dikarenakan terdapat duplikat data maka kita dapat menghapus baris tersebut.

Dengan demikian, data sudah siap untuk divisualisasikan. Dari *dataframe* tersebut, dibentuk beberapa visualisasi data sebagai berikut.

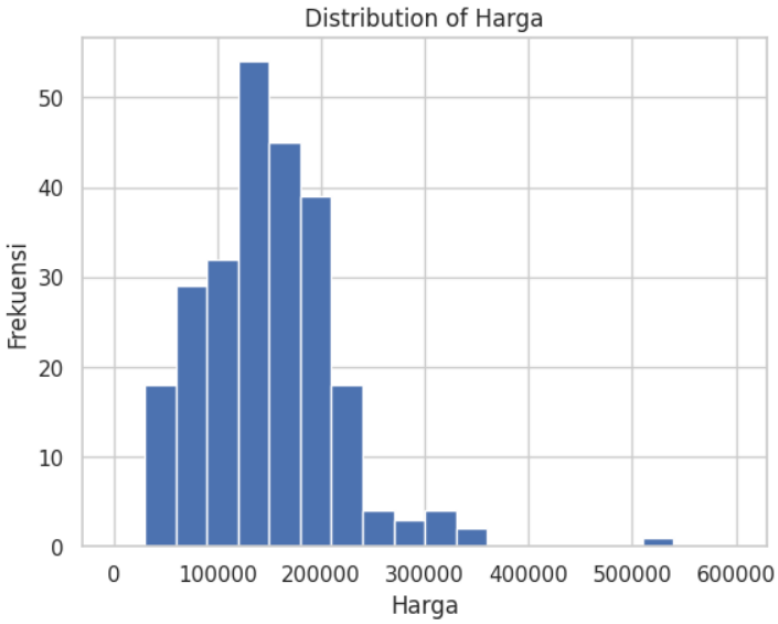
1. Histogram untuk melihat distribusi variabel “harga”, “rating”, dan “review”.
2. *Boxplot* untuk melihat distribusi variabel “harga”, “rating”, dan “review” secara lebih detail serta melihat apakah terdapat *outlier* pada variabel tersebut.
3. *Scatterplot* untuk mengetahui hubungan antara variabel “harga” dengan “rating”, “harga” dengan “review”, dan “rating” dengan “review”.
4. *Bar chart* untuk melihat jumlah produk dalam setiap kategori pada variabel “price info” dan pada variabel “banyak varian”.

Histogram

Histogram Variabel "Harga"

Histogram ini digunakan untuk mengetahui distribusi data harga dari *dataframe* tersebut. Dengan mengetahui distribusi harga, dapat dilihat *range* harga yang dominan karena histogram dapat menampilkan frekuensi untuk data harga tersebut.

Keluaran:



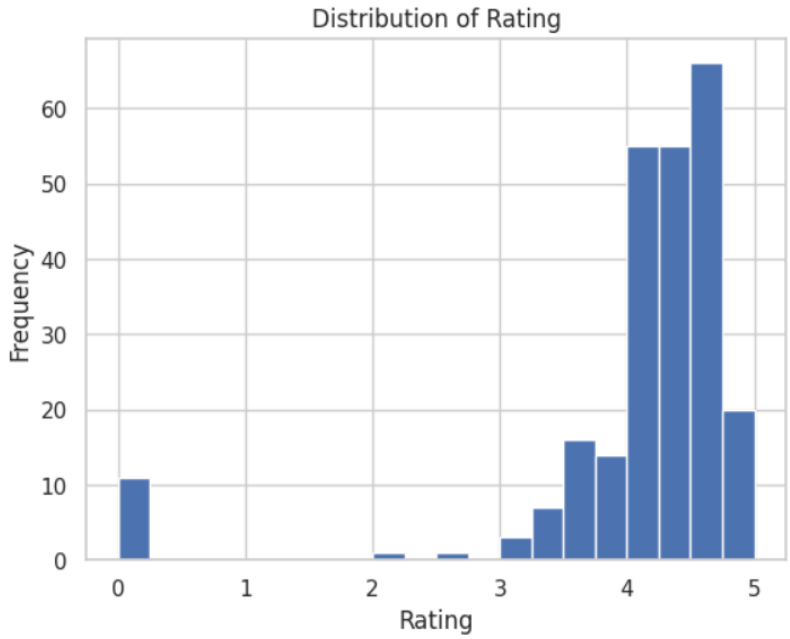
Interpretasi:

Dari histogram tersebut, dapat dilihat bahwa harga *lip product* di Sephora berkisar di antara 100.000 hingga 200.000, kemudian menurun secara perlahan-lahan pada rentang harga yang lebih tinggi. Dengan demikian, dapat disimpulkan bahwa mayoritas produk yang ada pada Sephora memiliki harga yang relatif terjangkau dan produk-produk dengan harga yang lebih tinggi jumlahnya lebih sedikit. Selain itu, dari histogram ini, dapat diketahui pula bahwa terdapat *outlier* pada data harga, yaitu terdapat produk dengan harga lebih dari Rp500.000.

Histogram Variabel "Rating"

Histogram ini digunakan untuk mengetahui distribusi data *rating* dari *dataframe* tersebut. Dengan mengetahui distribusi *rating*, kita dapat melihat *range rating* yang dominan karena histogram dapat menampilkan frekuensi untuk data *rating* tersebut.

Keluaran:



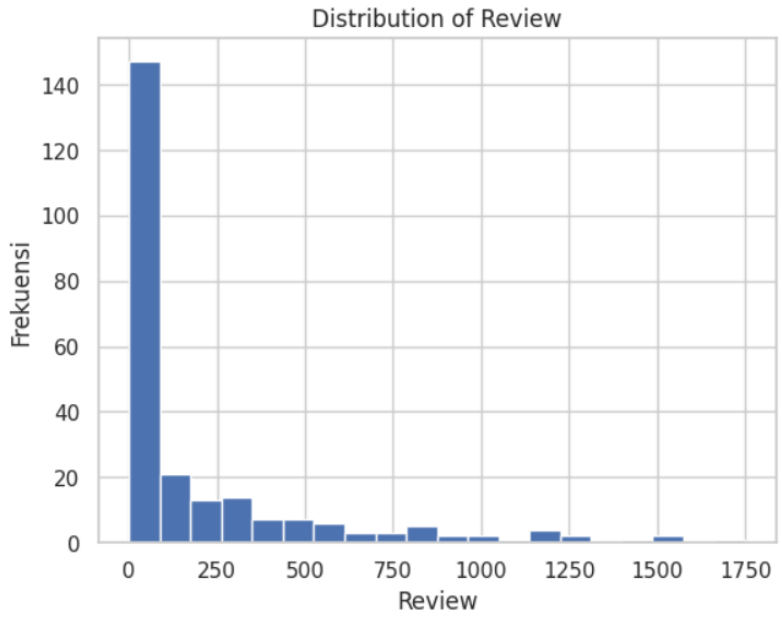
Interpretasi:

Dari histogram tersebut, dapat dilihat bahwa produk di Sephora memiliki *rating* yang berkisar antara 4 sampai dengan 5, kemudian menurun secara perlahan-lahan pada rentang rating yang lebih kecil. Dengan demikian, dapat kita simpulkan bahwa mayoritas produk yang ada pada Sephora memiliki rating yang cukup tinggi dan produk-produk dengan *rating* yang rendah jumlahnya lebih sedikit. Selain itu, dari histogram ini, dapat diketahui pula bahwa terdapat *outlier* pada data *rating*, yaitu terdapat produk-produk dengan rating kurang dari 1 atau 0.

Histogram Variabel "Review"

Histogram ini digunakan untuk mengetahui distribusi data *review* dari *dataframe* tersebut. Dengan mengetahui distribusi *review*, kita dapat melihat jumlah *review* yang dominan karena histogram dapat menampilkan frekuensi untuk data *review* tersebut.

Keluaran:



Interpretasi:

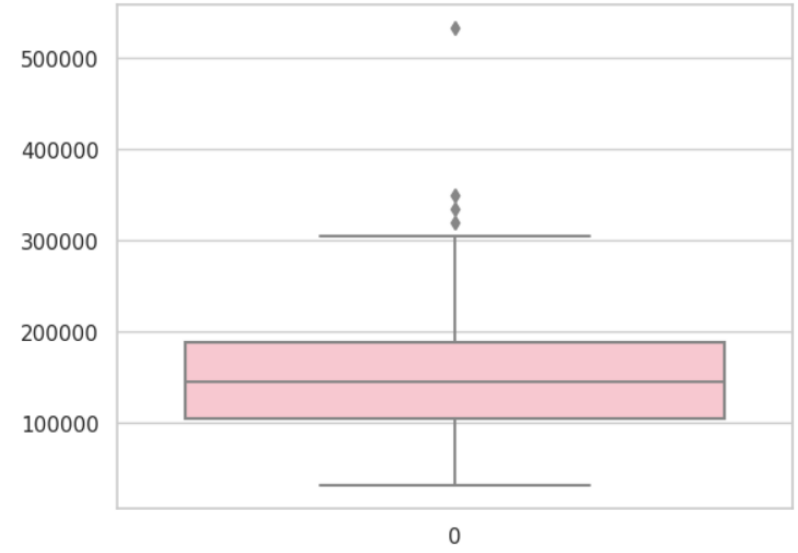
Dari histogram tersebut, dapat dilihat bahwa produk di Sephora memiliki jumlah *review* yang berkisar antara 0 hingga 500, kemudian menurun secara perlahan-lahan pada rentang jumlah *review* yang lebih tinggi. Dengan demikian, dapat disimpulkan bahwa mayoritas produk yang ada pada Sephora di-*review* oleh cukup banyak orang. Selain itu, dari histogram ini, dapat diketahui pula bahwa terdapat *outlier* pada data *review*, yaitu terdapat produk dengan jumlah review lebih dari 1.500.

Boxplot

***Boxplot* Variabel "Harga"**

Keluaran:

```
Quartile 1:  105000.0
Quartile 2:  145000.0
Quartile 3:  188000.0
Minimum value:  31200
Maximum value:  533000
```



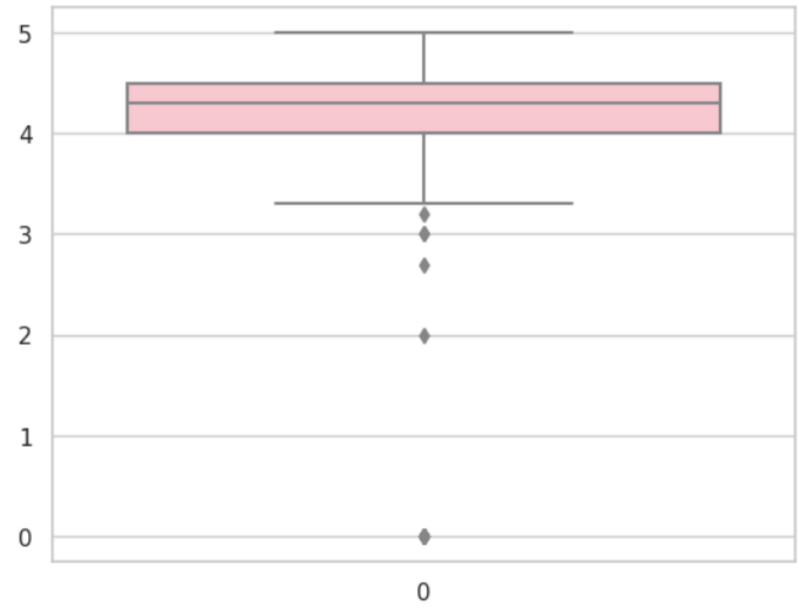
Interpretasi:

Berdasarkan *boxplot* di atas, dapat diketahui bahwa jarak antara garis horizontal tengah (Q_2) dengan Q_1 (40.000) lebih dekat daripada jarak Q_2 dengan Q_3 (43000). Dengan kata lain, harga tinggi relatif lebih menyebar daripada harga rendah dan keadaan ini disebut "menjurai ke atas". Kemudian, dapat diketahui pula bahwa kuartil bawah untuk harga produk adalah 105.000, median untuk harga produk adalah 145.000, kuartil atas untuk harga produk adalah 188.000, *minimum value* untuk harga produk adalah 31.200, dan *maximum value*-nya adalah 533.000. Dari *boxplot* tersebut, dapat diamati adanya *outlier* pada data harga produk.

***Boxplot* Variabel "Rating"**

Keluaran:

```
Quartile 1:  4.0
Quartile 2:  4.3
Quartile 3:  4.5
Minimum value:  0.0
Maximum value:  5.0
```



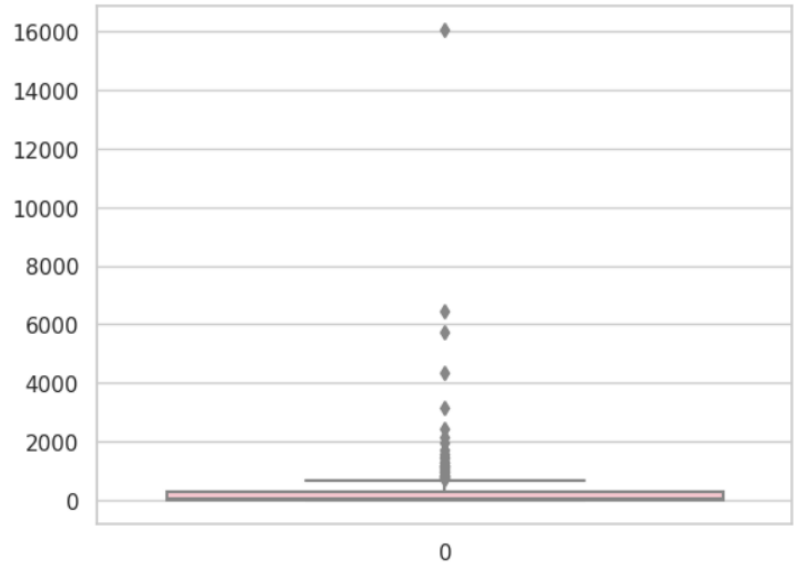
Interpretasi:

Berdasarkan *boxplot* di atas, dapat kita ketahui bahwa jarak antara garis horizontal tengah (Q_2) dengan Q_1 (0,3) lebih jauh daripada jarak Q_2 dengan Q_3 (0,2). Dengan kata lain, *rating* rendah relatif lebih menyebar daripada *rating* tinggi dan keadaan ini disebut "menjurai ke bawah". Kemudian, dapat diketahui pula bahwa kuartil bawah untuk *rating* produk adalah 4, median untuk *rating* produk adalah 4,3, kuartil atas untuk *rating* produk adalah 4,5, *minimum value* untuk *rating* produk adalah 0, dan *maximum value*-nya adalah 5. Dari *boxplot* tersebut, dapat diamati adanya *outlier* pada data *rating* produk.

***Boxplot* Variabel "Review"**

Keluaran:

```
Quartile 1:  8.0
Quartile 2:  46.0
Quartile 3:  298.0
Minimum value:  0
Maximum value:  16058
```



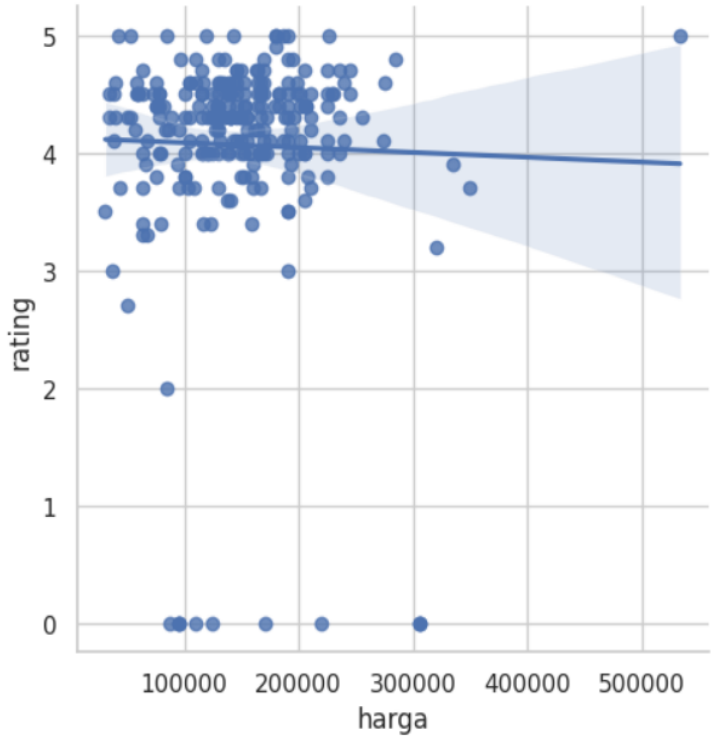
Interpretasi:

Berdasarkan *boxplot* di atas, dapat kita ketahui bahwa jarak antara garis horizontal tengah (Q_2) dengan Q_1 (38) lebih dekat daripada jarak Q_2 dengan Q_3 (252). Dengan kata lain, jumlah *review* banyak relatif lebih menyebar daripada jumlah *review* sedikit dan keadaan ini disebut "menjurai ke atas". Kemudian, dapat diketahui pula bahwa kuartil bawah dari jumlah *review* adalah 8, median untuk jumlah *review* adalah 46, kuartil atas untuk jumlah *review* adalah 298, *minimum value* untuk jumlah *review* produk adalah 0, dan *maximum value*-nya adalah 16.058. Dari *boxplot* tersebut, dapat diamati bahwa terdapat banyak *outlier* pada data jumlah *review* produk.

Scatterplot

***Scatterplot* untuk variabel "harga" dengan "rating"**

Keluaran:

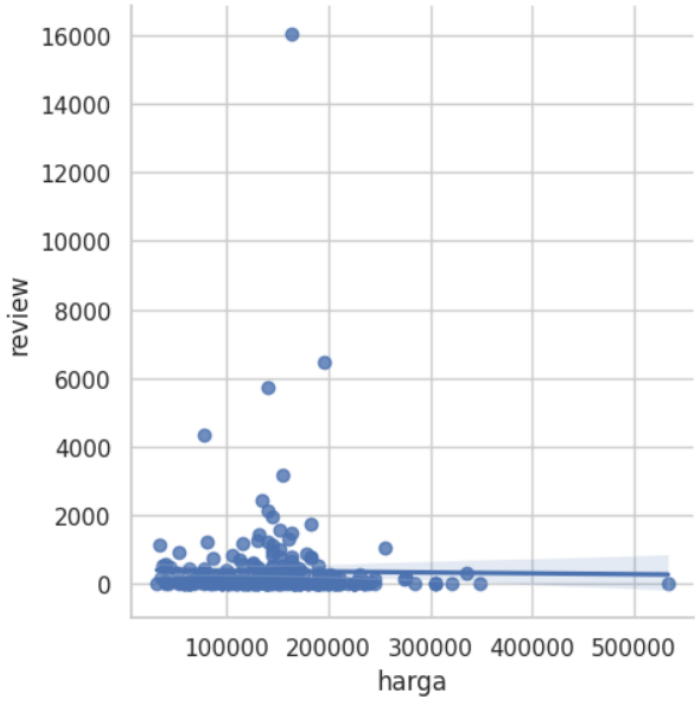


Interpretasi:

Berdasarkan *scatterplot* di atas, dapat diketahui bahwa terdapat hubungan antara variabel "harga" dengan "rating", di mana hubungan ini bersifat negatif. Artinya, semakin mahal harga jual suatu produk, semakin rendah *rating* produk tersebut.

***Scatterplot* untuk variabel "harga" dengan "review"**

Keluaran:

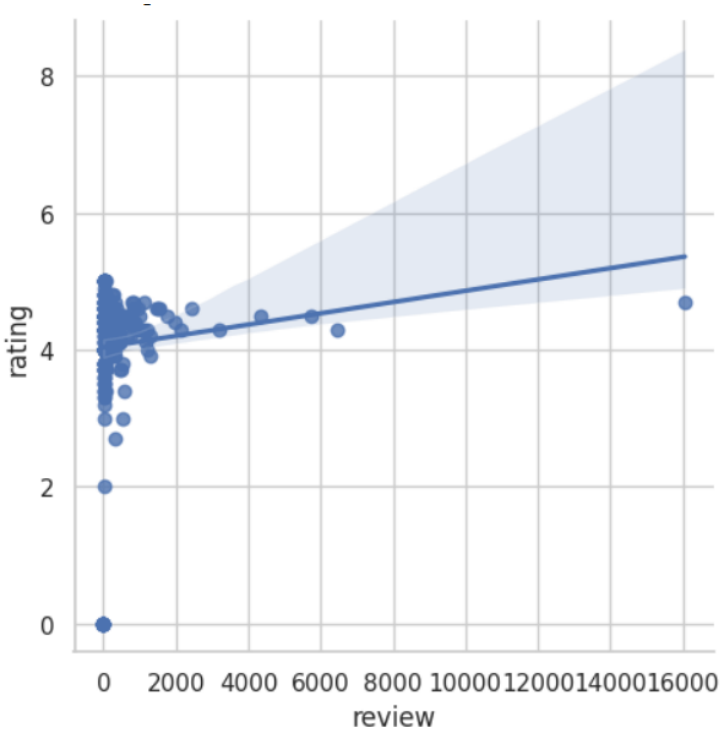


Interpretasi:

Berdasarkan *scatterplot* di atas, dapat diketahui bahwa tidak terdapat hubungan antara variabel "harga" dengan "review". Hal ini dikarenakan garis mendatar, sehingga dapat disimpulkan bahwa perubahan nilai pada pada variabel jumlah *review* tidak dipengaruhi oleh perubahan nilai pada variabel harga.

***Scatterplot* untuk variabel "review" dengan "rating"**

Keluaran:



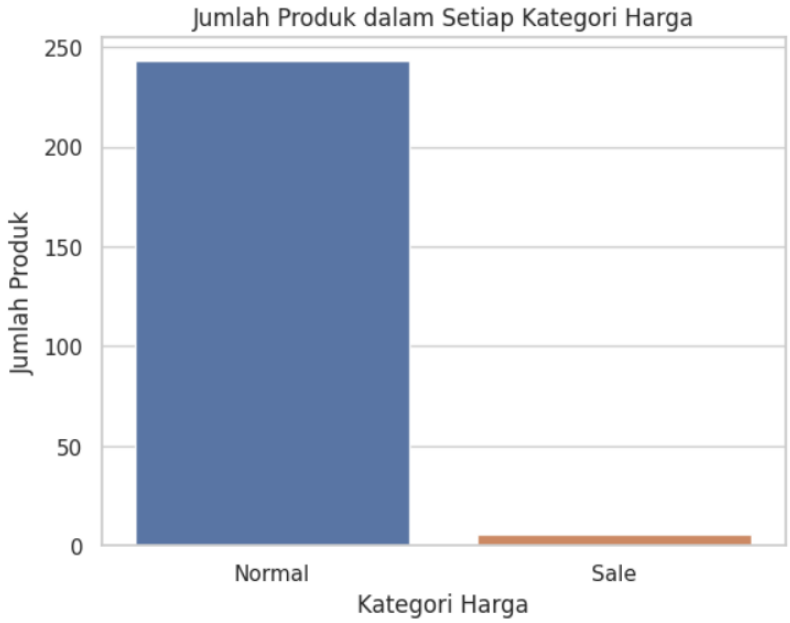
Interpretasi:

Berdasarkan *scatterplot* di atas, dapat diketahui bahwa terdapat hubungan antara variabel "review" dengan "rating", di mana hubungan ini bersifat positif. Artinya, semakin banyak orang yang me-*review* suatu produk, semakin tinggi pula *rating* produk tersebut.

Bar Chart

***Bar chart* untuk melihat jumlah produk dalam setiap kategori pada variabel 'price info'**

Keluaran:

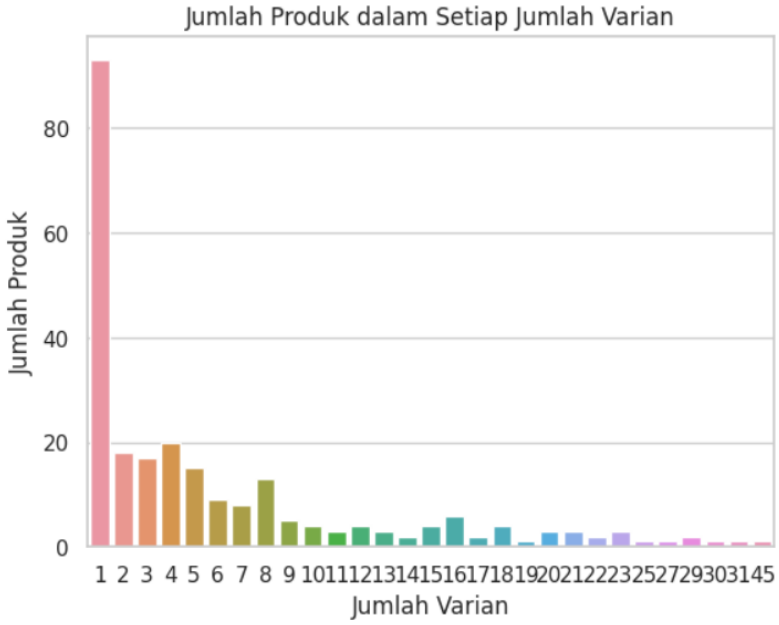


Interpretasi:

Berdasarkan *bar chart* di atas, dapat diketahui bahwa terdapat lebih dari 200 produk yang dijual dengan harga normal dan kurang dari 50 produk yang dijual dengan harga *sale*. Dengan demikian, dapat disimpulkan bahwa jumlah produk yang dijual dengan harga normal di Sephora jauh lebih banyak daripada jumlah produk yang dijual dengan harga *sale*.

***Bar chart* untuk melihat jumlah produk dalam setiap kategori pada variabel 'banyak varian'**

Keluaran:



Interpretasi:

Berdasarkan *bar chart* di atas, dapat diketahui bahwa terdapat lebih dari 80 produk yang dijual dengan hanya satu varian, 20 produk yang dijual dengan empat varian, dan untuk banyak varian lainnya, terdapat kurang dari 20 produk. Dengan demikian, dapat disimpulkan bahwa jumlah produk yang paling banyak dijual di Sephora hanya memiliki satu varian.

BAB III

PENUTUP

3.1 Kesimpulan

3.1.1 Kesimpulan Permasalahan Nomor 1

Sebelum melakukan *web scraping*, disiapkan terlebih dahulu *website* yang menjadi *target scraping*. Dipilih *link* https://www.webometrics.info/en/asia/indonesia%20, kemudian akan dilakukan *web scraping* dengan tahapan berpikir sebagai berikut.

1. Menentukan *website* yang akan di-*scraping* (dengan memperhatikan ketentuan apakah *website* tersebut boleh di-*scraping* atau tidak, informasi apa yang akan diambil, dan apakah informasi tersebut akan berguna atau tidak).
2. Menganalisis *website* tersebut untuk mengetahui struktur HTML dan URL-nya.
3. Menentukan *module* yang akan digunakan dengan menyesuaikan *website* yang akan di-*scraping*. Contoh *module* yang umum digunakan adalah `BeautifulSoup`, `Selenium`, `Scrapy`, dan `Requests`.
4. Membuat *code scraping* yang bisa membaca struktur HTML dari website tersebut dan mengambil data yang diinginkan. Kemudian, melakukan iterasi atau mengulang proses *scraping* pada seluruh halaman di *website* tersebut hingga diperoleh seluruh data.
5. Memeriksa kembali kesesuaian hasil *scraping*.

Tahapan *web scraping* adalah sebagai berikut.

1. Mengimpor *module* yang diperlukan.
2. Membuat *list* kosong untuk menyimpan hasil *scraping*.
3. Menyiapkan URL API dan mengecek statusnya terlebih dahulu. Apabila saat mengecek status URL dikembalikan *output* bernilai 200, dapat dilanjutkan dengan proses *scraping*.
4. Melakukan fungsi perulangan *for* untuk mengakses data universitas dari halaman pertama hingga halaman terakhir.
5. Menggunakan fungsi perulangan *for*, didapatkan atribut-atribut yang akan diakses dan dimasukkan ke dalam *array* atribut yang telah disiapkan.
6. Digunakan kembali fungsi *for* untuk mengambil elemen-elemen data yang diperlukan di dalam atribut. Dalam hal ini, diambil beberapa elemen data seperti *rank*, *world rank*, *name*, *impact*, *openness*, dan *excellence*. Setelah itu, setiap data dimasukkan ke dalam universitas.
7. Dibentuk *dataframe* menggunakan *module* pandas untuk memudahkan analisis dan visualisasi data.

3.1.2 Kesimpulan Permasalahan Nomor 2

Sebelum melakukan *API scraping*, disiapkan terlebih dahulu *website* yang menjadi *target scraping*. Pada tugas ini, digunakan *target website e-commerce*

Sephora, khususnya bagian *lip product* yang terdiri dari sekitar tujuh *page* dengan setiap *page* mengandung 30 hingga 36 produk kecantikan bibir. Tahapan API *scraping* yang dilakukan ialah sebagai berikut.

1. Mengimpor *module* yang diperlukan.
2. Menyiapkan URL API dan mengecek statusnya terlebih dahulu. Apabila saat mengecek status URL dikembalikan *output* bernilai 200, dapat dilanjutkan dengan proses *scraping*.
3. Membuat *array* kosong untuk menyimpan data-data pada *list* produk yang ditampilkan di *website*.
4. Melakukan fungsi perulangan *for* untuk mengakses produk dari halaman pertama hingga halaman terakhir.
5. Dari URL yang telah berhasil diakses, disiapkan *array* kosong lagi untuk menyimpan *list* atribut yang ada pada data di *website*.
6. Menggunakan fungsi perulangan *for*, didapatkan atribut-atribut yang akan diakses dan dimasukkan ke dalam *array* atribut yang telah disiapkan.
7. Digunakan kembali fungsi *for* untuk mengambil elemen-elemen data yang diperlukan di dalam atribut. Dalam hal ini, diambil beberapa elemen data seperti nama produk, harga, *rating*, *review*, info harga, dan lain-lain. Setelah itu, setiap data dimasukkan ke dalam *array* produk.
8. Karena *array* produk sudah terbentuk, dibentuk *dataframe* menggunakan *module* pandas untuk memudahkan analisis dan visualisasi data.

3.1.3 Kesimpulan Permasalahan Nomor 3

Berdasarkan *data frame* yang diperoleh dari nomor 1, dilakukan visualisasi data. Namun, sebelumnya, perlu dilakukan *data preprocessing*. Dari *data frame* universities, diperoleh bahwa terdapat 3.462 baris yang seluruhnya terisi dan tidak berduplikat. Akan tetapi, karena data numerik hendak divisualisasikan, beberapa variabel dengan tipe data *object* diubah menjadi *integer*. Setelah itu, diperoleh deskripsi numerik dari data. Dari visualisasi data terhadap *data frame* universities, diperoleh kesimpulan-kesimpulan sebagai berikut.

1. Dari *distribution plot*, diperoleh bahwa kebanyakan perguruan tinggi di Indonesia berada pada peringkat 30.000-an.
2. Dari *distribution plot*, diperoleh bahwa dampak eksternal kebanyakan perguruan tinggi di Indonesia berada pada peringkat 30.000-an.
3. Dari *distribution plot*, diperoleh bahwa keterbukaan kebanyakan perguruan tinggi di Indonesia berada pada peringkat 6.500-an.
4. Dari *distribution plot*, diperoleh bahwa kecemerlangan kebanyakan perguruan tinggi di Indonesia berada pada peringkat 7.000-an.
5. Dari *boxplot*, diperoleh bahwa distribusi data variabel “World Rank” menjurai ke bawah (*negative skewed*) dengan memuat sedikit pencilan bawah.
6. Dari *boxplot*, diperoleh bahwa distribusi data variabel “Impact” menjurai ke bawah (*negative skewed*) dengan memuat sedikit pencilan bawah.

7. Dari *boxplot*, diperoleh bahwa distribusi data variabel "Openness" menjurai ke bawah (*negative skewed*) dengan memuat hampir semua data sebagai pencilan bawah.
8. Dari *boxplot*, diperoleh bahwa distribusi data variabel "Excellence" menjurai ke bawah (*negative skewed*) dengan memuat hampir semua data sebagai pencilan bawah.
9. Dari *heatmap* korelasi, diperoleh bahwa korelasi terbesar terdapat antara variabel "World Rank" dengan "Impact", yaitu sebesar 0,98, sedangkan korelasi terkecil terdapat antara variabel "Excellence" dengan "Openness", yaitu sebesar 0,31.
10. Dari *bar chart*, diperoleh bahwa mayoritas (lebih dari tiga ribu) perguruan tinggi di Indonesia menempati peringkat dunia di atas 10.000.

Berdasarkan *data frame* yang diperoleh dari nomor 2, dilakukan visualisasi data. Namun, sebelumnya, perlu dilakukan *data preprocessing*. Dalam *data frame* tersebut, terdapat 44 *missing value* pada variabel "isi" dan tipe data untuk variabel "isi" belum sesuai. Akan tetapi, dikarenakan variabel "isi" tidak akan digunakan pada analisis, variabel "isi" dihapus dari *data frame*. Kemudian, diketahui terdapat data berduplikat, sehingga data berduplikat tersebut dihilangkan dahulu. Lalu, dilakukan beberapa visualisasi data dan diperoleh kesimpulan-kesimpulan sebagai berikut.

1. Dari histogram variabel "harga", dapat diketahui bahwa mayoritas produk yang ada pada Sephora memiliki harga yang relatif terjangkau. Meskipun demikian, terdapat produk yang harganya lebih dari Rp500.000.
2. Dari histogram variabel "rating", dapat diketahui bahwa mayoritas produk yang ada pada Sephora memiliki *rating* yang cukup tinggi. Meskipun demikian, terdapat produk yang *rating*-nya kurang dari 1.
3. Dari histogram variabel "review", dapat diketahui bahwa mayoritas produk yang ada pada Sephora di-*review* oleh cukup banyak orang (< 250 orang). Meskipun demikian, terdapat produk yang jumlah *review*-nya lebih dari 1.500 orang.
4. Dari *boxplot* variabel "harga", dapat diketahui bahwa harga tinggi relatif lebih menyebar daripada harga rendah serta terdapat *outlier* pada data harga produk. Kemudian, diketahui pula bahwa Q_1, Q_2, Q_3, nilai minimum, dan nilai maksimum dari harga produk secara berturut-turut adalah 105.000, 145.000, 188.000, 31.200, dan 533.000.
5. Dari *boxplot* variabel "rating", dapat diketahui bahwa *rating* rendah relatif lebih menyebar daripada *rating* tinggi serta terdapat *outlier* pada data *rating* produk. Kemudian, diketahui pula bahwa Q_1, Q_2, Q_3, nilai minimum, dan nilai maksimum untuk *rating* produk secara berturut-turut adalah 4, 4,3, 4,5, 0, dan 5.
6. Dari *boxplot* variabel "review", dapat diketahui bahwa jumlah *review* banyak relatif lebih menyebar daripada jumlah *review* sedikit serta terdapat banyak *outlier* pada data *review* produk. Kemudian, diketahui pula bahwa

Q_1, Q_2, Q_3, nilai minimum, dan nilai maksimum dari jumlah *review* produk secara berturut-turut adalah 8, 46, 298, 0, dan 16.058.

7. Dari *scatterplot* antara variabel "harga" dengan variabel "rating", dapat diketahui bahwa terdapat hubungan yang bersifat negatif antara kedua variabel ini. Artinya, semakin mahal harga jual suatu produk, semakin rendah *rating* produk tersebut.
8. Dari *scatterplot* antara variabel "harga" dengan variabel "review", dapat diketahui bahwa tidak terdapat hubungan antara kedua variabel ini. Artinya, perubahan nilai pada pada variabel jumlah *review* tidak dipengaruhi oleh perubahan nilai pada variabel harga.
9. Dari *scatterplot* antara variabel "review" dengan variabel "rating", dapat diketahui bahwa terdapat hubungan yang bersifat positif antara kedua variabel ini. Artinya, semakin banyak orang yang me-*review* suatu produk, semakin tinggi pula *rating* produk tersebut.
10. Dari *bar chart* jumlah produk pada setiap kategori variabel "price info", dapat diketahui bahwa jumlah produk yang dijual dengan harga normal di Sephora jauh lebih banyak daripada jumlah produk yang dijual dengan harga *sale*.
11. Dari *bar chart* jumlah produk pada setiap kategori variabel "banyak varian", dapat diketahui bahwa jumlah produk yang paling banyak dijual di Sephora hanya memiliki satu varian.

3.2 Kritik

Asisten-asisten praktikum MMS2434 Komputasi Statistika II tepat waktu dalam membagikan modul setiap minggu. Akan tetapi, terkadang asisten-asisten praktikum terlalu cepat dalam menjelaskan materi. Meski demikian, pembelajaran direncanakan dengan terstruktur dan bobot permasalahan-permasalahan yang diberikan dalam laporan praktikum UTS ini sudah sesuai dengan materi yang dipaparkan pada pertemuan-pertemuan praktikum.

3.3 Saran

Hendaknya, metode dan persiapan mengajar yang sudah dipraktikkan selama ini dapat dipertahankan sampai dengan akhir periode asistensi praktikum.