Proyek Data: Pengembangan Hotel Search Engine untuk DQLab Data

1. Approach dan Model yang Digunakan

Untuk ketiga kasus di dalam challenge DQlab kali ini, penulis menggunakan 2 model yang berbeda :

1. Untuk kasus pertama, menggunakan ***model recommender sederhana*** yang berbasis pada weighted rating atau berdasarkan rating dari tiap hotel tersebut
2. Untuk kasus kedua dan ketiga menggunakan ***content-based recommender*** yang berdasarkan dari fitur/informasi dari tiap hotel yang berbeda

2. ETL dan Data Cleansing

Data cleansing dilakukan kepada dataset untuk menghilangkan missing value /Nan serta merapikan nilai dari kolom price_per_night. Crawling data juga dilakukan untuk mendapatkan data hotel_city yang lebih akurat,karena masih ada hotel yang memiliki data hotel_city sebagai Jakarta saja.

Secara umum Langkah ETL dan data cleansing yang saya lakukan di antara lain :

1. Import library yang dibutuhkan
2. Load file dataset data_hotel dan review_hotel yang diberikan ke dataframe terpisah
3. Menghapus row data dengan missing value di masing-masing dataset
4. Menghapus nilai desimal dan membulatkan angka price_per_night ke angka puluhan terdekat
5. Crawling data hotel yang berada di Jakarta dari situs pegipegi.com untuk mendapat posisi hotel dengan data hotel_city = Jakarta yang lebih akurat

2.1 Import library yang dibutuhkan

In [40]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from matplotlib.colors import Normalize
from numpy.random import rand
import nltk
%matplotlib inline
```

2.2 Load dataset yang diberikan

In [3]:

```
hotel = pd.read_excel("data_hotel.xlsx")
review = pd.read_excel("review_hotel.xlsx")
```

2.3 Cleansing dataset

2.3.1 Cleansing dataset hotel

In [5]:

```
hotel.head()
```

Out[5]:

	hotel_id	hotel_name	hotel_description	hotel_province	hotel_city	hotel_address	price_per_night
0	h0001	Midtown Residence Marvell City Surabaya	NaN	Jawa Timur	Surabaya	Jalan Ngagel Raya No 123	527866.666667
1	h0002	favehotel Graha Agung Surabaya	Sebuah Hotel Mewah di Surabaya Persembahan dar...	Jawa Timur	Surabaya	Jl. Mayjen Yono Soewoyo Pakuwon Indah Square A...	442860.000000
2	h0003	The Sun Hotel Sidoarjo	Hotel Bintang 3 Pertama dan Satu-satunya di Si...	Jawa Timur	Surabaya	Jl. Pahlawan No.1,Sidokumpul, Sidoarjo, Suraba...	305000.000000
3	h0004	Grand Surabaya Hotel	Penginapan Yang Tenang Dan Nyaman Di Surabaya.	Jawa Timur	Surabaya	Jl. Pemuda 19-21, Surabaya, Indonesia	324999.333333
4	h0005	The WIN Hotel Surabaya	WIN Hotel adalah hotel smart bintang 3 yang me...	Jawa Timur	Surabaya	Jl. Embong Tanjung 46 - 48 Surabaya, Jawa Timu...	310947.250000

Terlihat bahwa di dataset hotel terdapat missing value di kolom description dan juga price per night. Juga kolom price_per_night masih memiliki angka desimal. Kita akan isi nilai null di kolom price dengan mean dari seluruh kolom dan menghapus data dengan missing value di kolom description

In [6]:

```
# menghapus string hotel_ di nama kolom
colname = [col.replace('hotel_','') for col in hotel.columns]
hotel.columns = colname

# mengisi nilai null di kolom price per night dengan nilai mean, dan melakukan pembulatan ke nilai puluhan terdek
at
hotel['price_per_night'] = hotel['price_per_night'].fillna(hotel.price_per_night.mean())
hotel['price_per_night'] = hotel['price_per_night'].apply(lambda x : int(round(x,-1)))

# menghapus data dengan nilai null di kolom description
hotel = hotel.dropna(subset=['description'])
```

2.3.2 Mendapatkan lokasi kota hotel yang lebih akurat

Beberapa hotel yang berada di provinsi DKI Jakarta memiliki data kota Jakarta saja. Untuk mendapat data kota yang lebih tepat, maka saya melakukan crawling data ke situs pegi2 untuk menarik informasi tersebut. Data yang dikumpulkan disimpan ke dalam satu file bernama "Hotel Name.csv" dan digabungkan dengan data hotel. (kode python untuk crawling data ini dapat dilihat melalui link berikut https://gist.github.com/davidsirait/bdb95661e101b4897bf2d3c73be432c6 (https://gist.github.com/davidsirait/bdb95661e101b4897bf2d3c73be432c6))

In [8]:

```
# load data nama-nama hotel dan lokasi hasil crawling
nama_hotel = pd.read_csv("Hotel Name.csv",index_col=0)
nama_hotel.head()
```

Out[8]:

	Hotel Name	Kota
0	Kyriad Hotel Metro Kebayoran	Jakarta Selatan
1	Artotel Jakarta Thamrin	Jakarta Pusat
2	Cabin Hotel	Jakarta Utara
3	Mega Matra Hotel	Jakarta Timur
4	Park 5 Simatupang	Jakarta Selatan

Selanjutnya adalah menggabungkan dan mengganti data lokasi kota tersebut ke dataframe hotel di awal

In [9]:

```
# merubah nama kolom di dataframe nama_hotel
nama_hotel = nama_hotel.rename(columns={"Hotel Name":"name"})

#menggabungkan kedua dataframe dan mengganti lokasi kota hotel yang sesuai
hotel = pd.merge(hotel,nama_hotel,how='left',on='name')
hotel.loc[hotel['Kota'].notnull(),'city'] = hotel['Kota']
hotel.loc[hotel.city == 'Jakarta'].head()
```

Out[9]:

	id	name	description	province	city	address	price_per_night	Kota
381	h0483	Zen Boutique Syariah Hotel	Pilihan akomodasi yang ideal untuk Pasangan, B...	DKI Jakarta	Jakarta	Jl. Kramat 6 No. 28 Rt. 001 Rw. 02 Kel., RT.2/...	143070	NaN
396	h0498	Clay Hotel Jakarta	Tempat yang tepat untuk wisatawan yang pintar\...	DKI Jakarta	Jakarta	JL. Blora No. 20, Thamrin, Jakarta	353400	NaN
421	h0523	Hotel Santika Kelapa Gading Jakarta	Sebuah kemewahan Group Santika di Kelapa Gadin...	DKI Jakarta	Jakarta	Mahaka Square Jl Kelapa Nias Raya Blok HF 3, K...	668330	NaN
423	h0525	Mercure Jakarta Gatot Subroto	-	DKI Jakarta	Jakarta	Jalan Gatot Subroto Kav. 1	670050	NaN
428	h0530	Hotel Belvena	Penginapan yang Nyaman di daerah Mangga Besar,...	DKI Jakarta	Jakarta	Jl. Mangga Besar No.49 A, Taman Sari, Jakarta,...	200000	NaN

Ternyata masih ada beberapa hotel yang nilai city nya masih Jakarta, yang berarti tidak berada di dalam data hasil crawling. Namun jika dilihat, rata-rata berada di kota Jakarta Pusat ataupun Jakarta Selatan. Oleh karena itu, value city akan diisi sebagian dengan Jakarta Pusat dan Jakarta Selatan

In [10]:

```
# mengganti data lokasi kota yang masih bernilai Jakarta dengan Jakarta Pusat dan Jakarta Selatan
hotel.loc[hotel[(hotel.city == 'Jakarta')].head(12).index,'city'] = 'Jakarta Pusat'
hotel.loc[hotel[(hotel.city == 'Jakarta')].head(12).index,'city'] = 'Jakarta Selatan'

# drop kolom Kota dari dataframe hotel
hotel = hotel.drop(columns='Kota')
```

2.3.3 Cleansing dataset review

In [11]:

```
review.head()
```

Out[11]:

	booking_id	booking_date	hotel_id	hotel_name	stay_duration	adults	children	rating	review
0	b0001	19-04-2020	h0014	Zest Hotel Jemursari Surabaya	1	2	1	8.4	Short stay
1	b0002	06-04-2020	h0014	Zest Hotel Jemursari Surabaya	1	1	1	10.0	Hotelnya nyaman
2	b0003	24-03-2020	h0014	Zest Hotel Jemursari Surabaya	2	2	1	9.2	Cukup baik untuk transit
3	b0004	23-03-2020	h0014	Zest Hotel Jemursari Surabaya	1	2	0	9.2	Nyaman
4	b0005	14-03-2020	h0014	Zest Hotel Jemursari Surabaya	2	2	1	6.8	Not good

In [12]:

```
# melihat berapa jumlah data dengan nilai NULL
review.isnull().sum()
```

Out[12]:

```
booking_id         0
booking_date       0
hotel_id         473
hotel_name         0
stay_duration      0
adults             0
children           0
rating             0
review             0
dtype: int64
```

Untuk dataset review, terdapat 473 row data yang memiliki nilai NULL di kolom hotel_id dan akan di drop

In [13]:

```
# hapus string hotel_ di nama kolom di dalam dataset
colname = [col.replace('hotel_','') for col in review.columns]
review.columns = colname

# drop row data yang memiliki nilai NULL
review = review.dropna()
```

3. Apakah Rating Penting?

Untuk case study pertama, pemodelan akan menggunakan simple recommender engine yang memberikan rekomendasi berdasarkan rating yang dimiliki hotel tersebut. Namun, tidak semua hotel memiliki nilai setara, karena tiap hotel memiliki count/jumlah tamu yang memberikan rating yang berbeda-beda, seperti ditunjukkan di bawah:

In [14]:

```
review_rating = review.groupby('name').agg({'id':'size','rating':'mean'}).rename(columns={'id':'count',}).reset_i
ndex()
review_rating.sort_values(by='count',ascending = False)
```

Out[14]:

	name	count	rating
163	Hotel Cemerlang	20	8.28
209	InterContinental Bandung Dago Pakar	20	9.52
365	Valore Hotel	20	7.48
359	U Janevalla Bandung	20	8.28
367	Verona Palace Hotel	20	8.32
...	...	...	...
97	Front One Residence Syariah Mampang	1	10.00
84	Ethan Hotel	1	10.00
50	Barito Mansion	1	9.60
20	Amaris Hotel Slipi	1	8.40
204	Hotel Zia Sanno Jakarta - Pluit	1	9.60

408 rows × 3 columns

Terlihat bahwa beberapa hotel memiliki jumlah vote (count) sebanyak 20, artinya terdapat 20 user yang memberikan nilai ke hotel tersebut. Namun ada juga hotel yang memiliki hanya 1 vote saja, sehingga ada kemungkinan hotel yang secara umum lebih populer dan dituju banyak orang ratingnya berada di bawah hotel yang hanya disukai segelintir orang saja . Sebagai contoh dari data di atas, InterContinental Bandung Dago Pakar dengan rating 9,52 dari 20 vote jika diurutkan berdasarkan rating saja akan berada di bawah Ethan Hotel yang memiliki rating 10, namun berasal dari 1 vote.

3.1 Mengolah data rating untuk dimasukkan ke model

Mengikuti referensi simple recommender engine, maka pemodelan dapat menggunakan rumus Weighted Rating IMDB yaitu :

$$WR = \frac{v}{v+m} \cdot R + \frac{m}{v+m} \cdot C$$

Dimana :
v adalah jumlah vote (count) yang diterima hotel tersebut
m jumlah minimum vote yang dibutuhkan agar dapat masuk ke dalam perhitungan
R adalah rating rata-rata dari hotel itu sendiri ,dan
C adalah rata-rata dari **keseluruhan hotel** di dalam dataset

v dan R sudah terdapat di dalam dataset review_rating,sementara nilai C dapat kita cari menggunakan metoda mean() terhadap dataset review, dan R didapat berdasarkan data count dari berapa kali hotel muncul di dataset yang ada di dalam dataset. Metode Weighted Rating ini dirasa secara matematis cukup sederhana serta memperhitungkan rating serta jumlah vote yang didapat hotel tersebut secara lebih adil.

3.2 Nilai rating untuk pemodelan content-based recommender

Pemodelan content-based menggunakan deskripsi/informasi dari beberapa fitur untuk merekomendasikan item yang memiliki kemiripan dengan informasi yang digunakan sebelumnya. Metode ini pada umumnya menggunakan konsep vector space model, yang intinya merubah text ke dalam representasi aljabar berbentuk vektor. Sehingga, nilai rating yang berbentuk integer **tidak terlalu berpengaruh** untuk pemodelan ini karena lebih berfokus pada data implisit seperti teks

4. EDA

4.1 Hotel Analysis

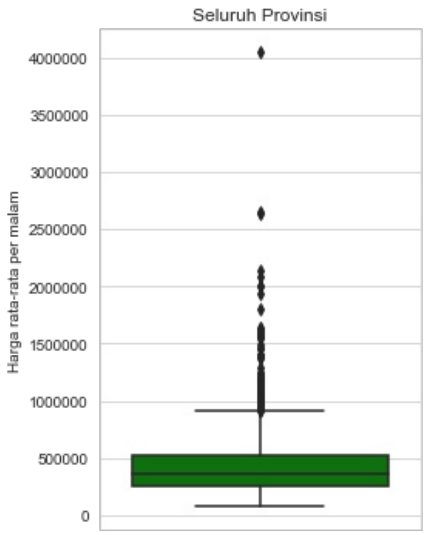
Langkah ini dimulai dengan melakukan analisis dari data yang terdapat di dalam dataframe hotel

4.1.1 Harga hotel berdasarkan provinsi

Pertama-tama kita lihat dapat melihat distribusi dari harga kamar tiap hotel di seluruh daerah menggunakan box plot

In [16]:

```
# box plot untuk melihat distribusi dari harga rata-rata per malam seluruh hotel
plt.figure(1,figsize = (4,5))
sns.set_style('whitegrid')
sns.boxplot(hotel['price_per_night'],color='green',orient='v')
plt.title('Seluruh Provinsi')
plt.ylabel('Harga rata-rata per malam')
plt.tight_layout()
```



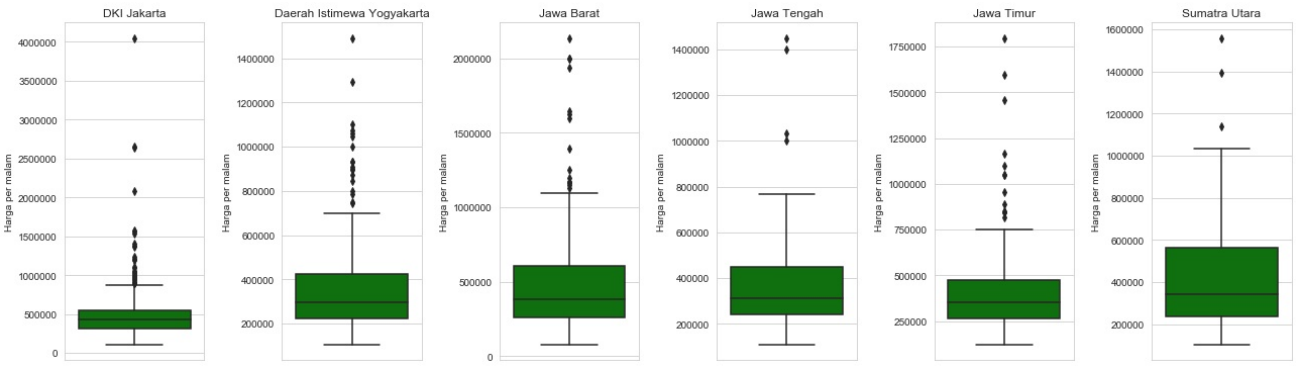
Terlihat bahwa harga rata-rata maksimum yang tercatat adalah sekitar 4 juta rupiah per malam. Nilai ini jauh berada di atas quartile ke -3 (75%) yang berada di angka sekitar 520 ribu rupiah. Sementara itu rata-rata hotel di seluruh daerah mematok tarif per malam sekitar 460 ribu rupiah, dan sekitar 50 % dari total seluruh hotel berada di sekitar angka ini. Dapat disimpulkan bahwa sebagian besar hotel memasang tarif yang cukup terjangkau untuk bersaing, meskipun ada juga hotel yang berada sebagai outlier dan memiliki tarif di atas 1 juta per malamnya

Untuk mendapat gambaran lebih jelas kita bisa melihat bagaimana persebaran harga hotel per malam berdasarkan provinsi dan juga kota, dengan cara melakukan grouping per dan di plotting dengan box plot

In [17]:

```
# grouping berdasarkan provinsi
hotel_province = hotel[['province','price_per_night']]
hotel_province = hotel_province.groupby('province')['price_per_night'].apply(list).reset_index(name = 'harga')

#plotting boxplot untuk distribusi harga per malam
data = hotel_province['harga']
number_of_columns=6
number_of_rows = len(data)-1/number_of_columns
plt.figure(figsize=(3*number_of_columns,5*number_of_rows))
for i in range(0,len(data)):
    plt.subplot(number_of_rows + 1,number_of_columns,i+1)
    sns.set_style('whitegrid')
    sns.boxplot(data[i],color='green',orient='v')
    plt.title(hotel_province['province'][i])
    plt.ylabel('Harga per malam')
    plt.tight_layout()
```



Dari grafik terlihat bahwa Jakarta memiliki tarif rata-rata yang paling tinggi di antara semua daerah, dan juga hotel kelas atas yang harga per malamnya berada di atas 2 juta per malam. Sumatra Utara memiliki range harga yang paling tinggi, terlihat dari cakupan box hijau yang berada di antara 220 - 580 ribu. Namun, hanya sedikit hotel di Sumatra yang memiliki tarif jauh di luar rata-rata. Sementara,harga kamar hotel di DI yogyakarta secara umum persebarannya lebih rendah dibandingkan daerah lain, disusul oleh Jawa Tengah. Namun, di Yogyakarta masih banyak hotel yang harga per malamnya dapat dianggap jauh di atas rata-rata(outlier)

4.1.2 Harga hotel berdasarkan kota

Berikutnya kita akan membandingkan persebaran tarif kamar per malam berdasarkan lokasi kota tempat hotel berada

In [19]:

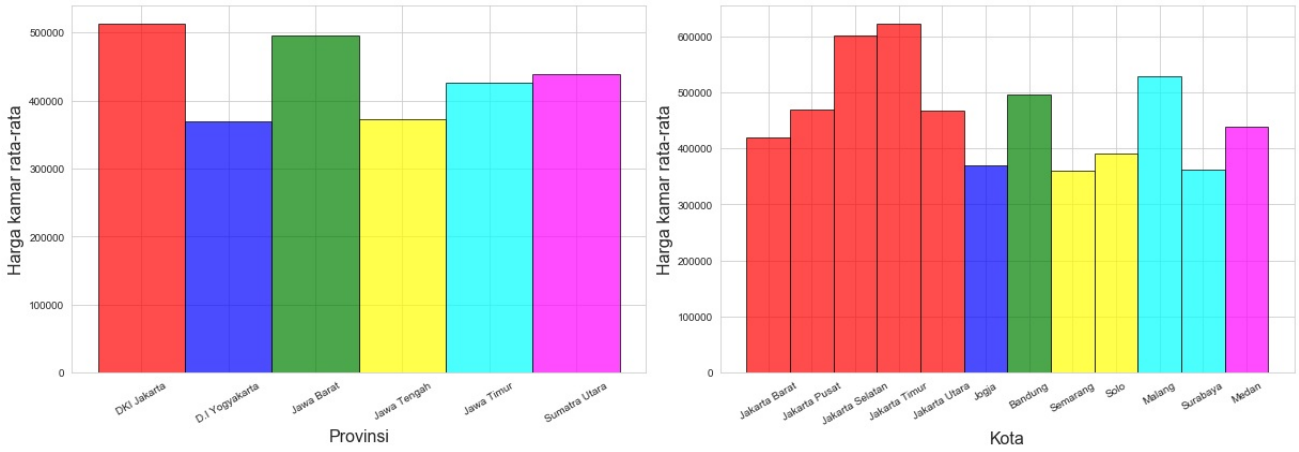
```
# grouping kembali hotel berdasarkan provinsi untuk dibandingkan dengan berdasarkan kota
hotel_grup_province = pd.DataFrame(hotel.groupby('province').agg({'id':'size','price_per_night':'mean'})
                                   .rename(columns={'id':'count',}).reset_index()).replace('Daerah Istimewa Yogya
karta','D.I Yogyakarta')

# grouping berdasarkan kota dan cari harga rata-rata kamar per malam
hotel_grup_city = pd.DataFrame(hotel.groupby('city').agg({'id':'size','price_per_night':'mean',
                                                          'province':lambda x: x.unique().tolist()})
                                   .rename(columns={'id':'count',}).reset_index())
hotel_grup_city = hotel_grup_city.sort_values(by ='province',ascending = True)
```

In [20]:

```
# plot harga kamar rata-rata berdasarkan provinsi dan juga kota
plt.figure(1,figsize = (17,6))
plt.subplot(121)
plt.bar(hotel_grup_province['province'], hotel_grup_province['price_per_night'],
        label = 'Harga kamar rata-rata per provinsi', color = ['red','blue','green','yellow','cyan','magenta'],
        width = 1, align = 'center', alpha = 0.7,edgecolor = 'black')
plt.xlabel('Provinsi', fontsize = 16)
plt.ylabel('Harga kamar rata-rata', fontsize = 16)
plt.xticks(rotation=30)

plt.subplot(122)
plt.bar(hotel_grup_city['city'], hotel_grup_city['price_per_night'],
        label = 'Harga kamar rata-rata per kota', color = ['red','red','red','red','red','blue','green',
                                                           'yellow','yellow','cyan','cyan','magenta']
        ,  width = 1, align = 'center', alpha = 0.7,edgecolor = 'black')
plt.xlabel('Kota', fontsize = 16)
plt.ylabel('Harga kamar rata-rata', fontsize = 16)
plt.xticks(rotation=30)
plt.tight_layout()
```



Hotel di Jakarta Timur memiliki tarif rata-rata yang paling tinggi, disusul hotel di daerah Jakarta Selatan yang notabene memiliki beberapa daerah bisnis dan juga perbelanjaan. Sementara, terlihat bahwa kota di kota Malang tarif hotel per malamnya relatif lebih tinggi dibanding daerah-daerah lainnya dan cukup kontras dengan Surabaya yang merupakan ibukota provinsi Jawa Timur. Sementara kota Jogja, sebagai tujuan destinasi wisata memiliki tarif rata-rata yang paling rendah di antara kota-kota lainnya. Maka jika ingin melakukan perjalanan wisata dengan budget terbatas, kota Jogja dapat menjadi pilihan

4.1.3 Data deskripsi Hotel

Berikutnya kita akan menganalisa kolom deskripsi untuk melihat bagaimana cara hotel memasarkan dirinya kepada calon pengunjung potensial

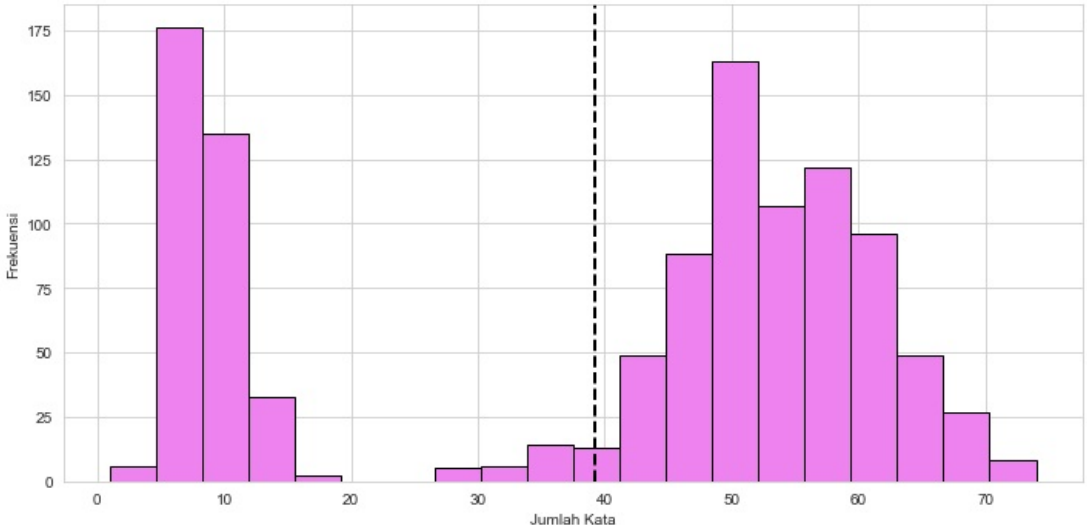
In [21]:

```
# membuat copy dari dataframe hotel untuk analisa kolom description
hotel_test = hotel.copy()
hotel_test['jumlah_kata'] = hotel_test['description'].apply(lambda x: len(str(x).split()))
desc_lengths = list(hotel_test['jumlah_kata'])
print("Jumlah hotel yang memiliki deskripsi:", len(desc_lengths),
    "\nJumlah kata rata-rata", np.average(desc_lengths),
      "\nJumlah kata maksimum", max(desc_lengths),
      "\nJumlah kata minimum", min(desc_lengths))
```

```
Jumlah hotel yang memiliki deskripsi: 1099
Jumlah kata rata-rata 39.21565059144677
Jumlah kata maksimum 74
Jumlah kata minimum 1
```

In [22]:

```
# Plot jumlah kata yang digunakan beserta frekuensinya
plt.figure(1,figsize = (10,5))
plt.hist(desc_lengths, density=False, bins=20,color = 'violet',edgecolor='black')
plt.axvline(np.average(desc_lengths), color='k', linestyle='dashed', linewidth=2)
plt.ylabel('Frekuensi')
plt.xlabel('Jumlah Kata');
plt.tight_layout()
plt.show()
```



Sebagian besar hotel sudah memberikan deskripsi yang cukup banyak, dilihat dari cukup banyak hotel yang panjang deskripsinya di atas rata-rata , yaitu 39 kata. Namun masih banya hotel yang hanya memberikan deskripsi singkat sebanyak 1- 10 kata saja, dan mungkin kurang menarik untuk para calon pengunjung.

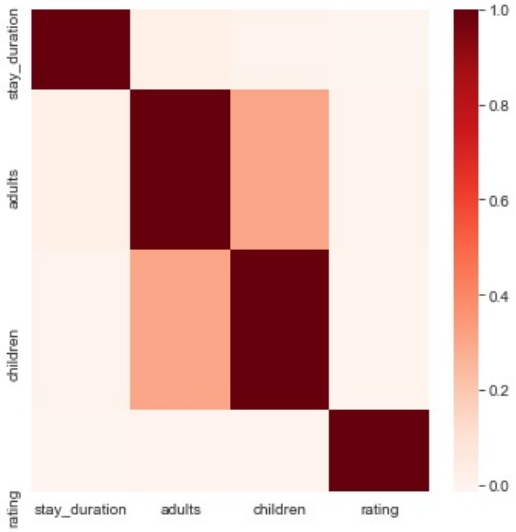
4.2 Review Analysis

4.2.1 Korelasi antara lama menginap, jumlah orang serta rating yang diberikan

Untuk analisa dataframe review, penulis ingin melihat apakah ada pengaruh yang signifikan antara variabel durasi menginap dan jumlah tamu terhadap rating yang diberikan untuk tiap hotel menggunakan heatmap dan pearson correlation

In [23]:

```
plt.figure(figsize=(5,5))
sns.heatmap(review.corr(),cmap='Reds',annot=False)
plt.tight_layout()
```



In [24]:

```
review.corr(method='pearson')
```

Out[24]:

	stay_duration	adults	children	rating
stay_duration	1.000000	0.020079	0.001333	-0.013479
adults	0.020079	1.000000	0.303652	0.001924
children	0.001333	0.303652	1.000000	-0.002190
rating	-0.013479	0.001924	-0.002190	1.000000

Terlihat dari grafik heatmap antara children, adults, serta stay_duratin dengan rating memiliki warna cerah, yang berarti tidak ada korelasi yang signifikan antara ketiga variabel dan rating. Tabel korelasi pearson juga menunjukkan angka korelasi ketiga variabel mendekati nol (kolom paling kanan). Hal ini berarti durasi menginap serta jumlah tamu yang menginap bisa kita abaikan saat membentuk model

4.2.2 Distribusi rating untuk tiap hotel

Untuk melihat distribusi rating tiap hotel, perlu dilakukan grouping berdasarkan nama hotel untuk mencari count dan average rating per hotel. Hal ini sudah dilakukan di bagian sebelumnya. Dengan describe kita dapat melihat informasi kuantitatif untuk tiap kolom di dataframe review, yang kemudian diplot ke dalam grafik

In [25]:

```
review_rating.describe()
```

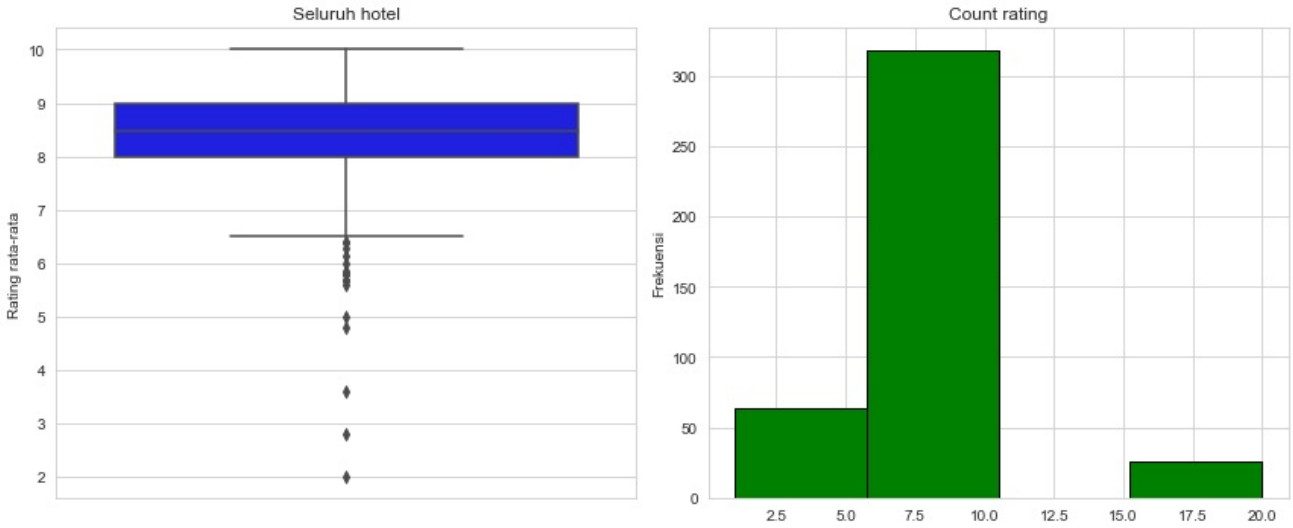
Out[25]:

	count	rating
count	408.000000	408.000000
mean	9.394608	8.371492
std	3.804003	0.976643
min	1.000000	2.000000
25%	10.000000	7.990000
50%	10.000000	8.480000
75%	10.000000	9.000000
max	20.000000	10.000000

In [26]:

```
# plotting
plt.figure(1,figsize = (12,5))
plt.subplot(121)
sns.set_style('whitegrid')
sns.boxplot(review_rating['rating'],color='blue',orient='v')
plt.title('Seluruh hotel')
plt.ylabel('Rating rata-rata')

plt.subplot(122)
plt.hist(review_rating['count'], density=False, bins=4,color = 'green',edgecolor='black')
plt.title('Count rating')
plt.ylabel('Frekuensi')
plt.tight_layout()
```



Rata-rata nilai rating untuk tiap hotel berada di angka 8.3, dengan standar deviasi 0.97. Artinya user memberikan rating yang cukup baik kepada masing masing hotel, dan rating dibawah angka 7 dapat dianggap sebagai outlier/sangat jarang terjadi. Jumlah vote/count untuk tiap hotel mayoritas berada di antara 6 sampai 10 jumlah vote

5. Fitur yang Digunakan

Fitur - fitur yang akan dimasukkan ke dalam pemodelan adalah :

1. Count dan rating : untuk kasus pertama menggunakan model simple recommender system
2. Hotel description : informasi mengenai tiap hotel akan menjadi basis untuk melihat kemiripan antara satu hotel dengan yang lainnya
3. Hotel province, city, dan address : informasi ini berguna untuk mengutamakan hotel yang berada di lokasi yang sama untuk direkomendasikan
4. Review : melihat similarity review antar hotel

6. Pembuatan Model

6.1 Kasus 1 : Simple Recommender

Langkah pertama adalah menggabungkan dataframe review_rating yang berisi informasi rating rata-rata tiap hotel dengan dataframe hotel menjadi dataframe dengan nama data_join

In [27]:

```
# join review_rating ke dataframe hotel
data_join = hotel.set_index('name').join(review_rating.set_index('name')).reset_index()
data_join = data_join.dropna()
```

Selanjutnya kita akan menentukan nilai m, yaitu jumlah minimum vote yang dibutuhkan. Mengacu pada referensi dari modul DQLab, digunakan m yaitu 80% dari data, atau dalam hal ini minimum 10 vote

In [29]:

```
# menghitung nilai m
m = data_join['count'].quantile(0.8)

# fungsi untuk menentukan Weighted Rating masing2 hotel
def weightedRating(df,var):
    v = df['count']
    R = df['rating']
    C = df['rating'].mean()
    m = df['count'].quantile(var)
    df['score'] = (v/(m+v))*R + (m/(m+v))*C #Rumus WR
    return df['score']

weightedRating(data_join,0.8)
print()
```

Terakhir kita membuat sistem rekomendasi yang dapat menerima input seperti lokasi kota dan provinsi dari user dan mengembalikan rekomendasi hotel berdasarkan rating terbaik

In [30]:

```
# model rekomendasi yang menampilkan rekomendasi 5 hotel terbaik

def simpleRecommender(df,province,city,top=5):
    # specified the province
    df = df[df['province'] == province]

    # specified the city
    df = df[df['city'] == city]

    # filter hotels with count > m
    df = df[df['count'] >= m]
    df = df.sort_values(by = 'score', ascending = False)

    # take top 5 hotels
    df = df[:top]
    return df
```

In [31]:

```
# mengurutkan hotel hasil rekomendasi berdasarkan harga terendah
simpleRecommender(data_join,'Jawa Timur','Surabaya').sort_values(by='price_per_night')
```

Out[31]:

	name	id	description	province	city	address	price_per_night	count	rating	score
200	Evora Hotel Surabaya	h0007	Ketika bisnis dan kenyamanan sinergi di satu t...	Jawa Timur	Surabaya	Jl. Menur 18 - 20, Surabaya, Indonesia	283000	10.0	9.04	8.704727
701	Neo+ Waru Sidoarjo by ASTON	h0010	Pilihan akomodasi yang ideal untuk Pasangan, B...	Jawa Timur	Surabaya	Jl. S. Parman No. 52-54, Waru, Sidoarjo, Surab...	293120	10.0	9.48	8.924727
101	Best Western Papilio Hotel	h0016	Welcome to Best Western Papilio Hotel!	Jawa Timur	Surabaya	Jl. Ahmad Yani 176 - 178, Surabaya	305890	10.0	9.52	8.944727
1093	favehotel Sidoarjo	h0011	Penginapan Yang Tenang Dan Nyaman di Sidoarjo.	Jawa Timur	Surabaya	Jl. Jenggolo No. 15, Pucang, Kec. Sidoarjo, Ka...	447220	10.0	9.08	8.724727
1054	Yello Hotel Jemursari	h0032	Sebuah hotel yang nyaman dan strategis di Sura...	Jawa Timur	Surabaya	Jl. Raya Jemursari 176, Surabaya, Jawa Timur, ...	513000	10.0	9.28	8.824727

6.2 Kasus 2 : Content-Based Filtering

Pada kasus ini, content based filtering menggabungkan informasi seperti deskripsi hotel, alamat, lokasi dan review ke dalam satu 'bag of words' yang akan diubah menjadi vektor matematis. Khusus untuk review, perlu dilakukan manipulasi untuk menggabungkan review yang bermacam-macam untuk tiap hotel sehingga menjadi satu kesatuan string sebelum digabungkan dengan fitur lainnya

In [32]:

```
# menggabungkan review untuk tiap hotel ke dalam satu string panjang
review_group = review.groupby('name')['review'].apply(list).reset_index()
review_group['review'] = review_group['review'].apply(lambda x : ",".join(x))
review_group.head()
```

Out[32]:

	name	review
0	45 Residence	Not recommended
1	7 Days Premium Hotel	Hotel terburuk yg pernah sy singgahi..,Nice ho...
2	AONE Hotel	Hotel bersih dan strategis,Lokasi hotel strate...
3	AYANA Midplaza Jakarta	Goodplace,Recommended,Recommended,Liburan kelu...
4	Adimulia Hotel Medan	Bagus,Hotelnya keren,Jw mariot,Hotelnya ckup m...

Selanjutnya adalah menggabungkan review untuk tiap hotel tersebut ke dalam dataframe hotel. Untuk hotel-hotel yang tidak memiliki data review, maka diset value reviewnya menjadi unknown

In [33]:

```
# join review dengan datset hotel, dan mengisi nilai review NULL dengan unknown
content_join = hotel.set_index('name').join(review_group.set_index('name')).reset_index()
content_join = content_join.fillna("unknown")
```

Berikutnya, sebelum menggabungkan semua fitur menjadi satu 'bag of words', kita akan 'membersihkan' teks di dalam fitur untuk menghilangkan tanda baca dari tiap fitur, merubah semua kata menjadi lowercase, serta menghilangkan stopwords. Bahasa Indonesia dan Bahasa Inggris terdapat di beberapa fitur, sehingga kita perlu menghilangkan stopwords untuk kedua bahasa. Stopwords bahasa Indonesia didapat menggunakan library sastrawi, sedangkan stopwords bahas inggris menggunakan library nltk

In [34]:

```
# import library Sastrawi,re dan nltk
import re
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
from nltk.corpus import stopwords

# set special character dan symbol yang ingin dihilangkan
clean_spcl = re.compile('[/(){}\[\]\|@,;] ')
clean_symbol = re.compile('[^0-9a-z #+_]')
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
stopworda = set(stopwords.words('english')) # set stopwords untuk bahasa inggris

feature = ['description','province','city','address','review'] # list berisi feature yang akan dibersihkan

# fungsi untuk membersihkan teks
def clean_text(text):
    text = str(text)
    text = text.lower() # lowercase text
    text = clean_spcl.sub(' ', text)
    text = clean_symbol.sub(' ', text)
    text = stopword.remove(text) # hapus stopword b. indonesia
    text = ' '.join(word for word in text.split() if word not in stopworda) # hapus stopword b.inggris
    return text

# apply fungsi ke tiap kolom fitur
for col in feature:
    content_join[col] = content_join[col].apply(clean_text)
```

Berikutnya kita akan menggabungkan semua fitur yang sudah dibersihkan ke dalam kolom baru bernama soup

In [35]:

```
# fungsi untuk menggabungkan kesemua fitur
def soup_feature(x):
    return ''.join(x['description']) + ' ' + ''.join(x['province']) + ' ' + ''.join(x['city']) + ' ' + ''.join(x[
'address']) + ' ' + ''.join(x['review'])

# membuat soup menjadi 1 kolom
content_join['soup'] = content_join.apply(soup_feature,axis=1)
```

Berikutnya kita akan melakukan vectorizer ke kolom soup dan menggubahnya ke dalam bentuk vektor. Kali ini algoritma yang digunakan adalah TF-IDF (Term Frequency-Inverse Document Frequency) dari library scikit learn. TF-IDF memberikan *weight* kepada kata yang terdapat di dalam teks,tergantung seberapa sering kata itu muncul. Hal ini karena beberapa kata, misalnya kata 'strategis' di kolom deskripsi, lebih sering muncul dibanding kata lain. Setelah itu kita bisa membuat matriks cosine similarity score untuk seluruh hotel

In [36]:

```
# import TF-IDF dan cosine similarity dari library sklearn
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# membuat tfidf vectorizer
tf = TfidfVectorizer()
tf_matrix = tf.fit_transform(content_join['soup'])

# membuat matriks similarity score untuk tfidf
tf_sim = cosine_similarity(tf_matrix,tf_matrix)
tf_sim
```

Out[36]:

```
array([[1.        , 0.10211544, 0.21270458, ..., 0.00341225, 0.01598582,
        0.01710495],
       [0.10211544, 1.        , 0.10244727, ..., 0.10776574, 0.        ,
        0.        ],
       [0.21270458, 0.10244727, 1.        , ..., 0.02175095, 0.0013481 ,
        0.00144247],
       ...,
       [0.00341225, 0.10776574, 0.02175095, ..., 1.        , 0.        ,
        0.        ],
       [0.01598582, 0.        , 0.0013481 , ..., 0.        , 1.        ,
        0.71654648],
       [0.01710495, 0.        , 0.00144247, ..., 0.        , 0.71654648,
        1.        ]])
```

Langkah terakhir adalah membuat model content-based recommender untuk menghasilkan rekomendasi hotel

In [37]:

```
# menjadikan nama hotel sebagai anchor index untuk input rekomendasi
indices = pd.Series(content_join.index, index=content_join['name']).drop_duplicates()

# base dataframe sebagai output rekomendasi
base = content_join.drop(columns=['review','soup'])

def content_recommender(hotel):
    # mendapatkan index dari hotel yang disebutkan
    idx = indices[hotel]

    #menjadikan list dari array similarity tf sim
    sim_scores = list(enumerate(tf_sim[idx]))

    #mengurutkan film dari similarity tertinggi ke terendah
    sim_scores = sorted(sim_scores,key=lambda x:x[1],reverse=True)

    #untuk mendapatkan list hotel dimulai dari indeks ke 1 sampai ke 6
    sim_scores = sim_scores[1:6]

    #mendapatkan index dari nama hotel yang muncul di sim_scores dan mengurutkan berdasarkan harga
    hotel_indices = [i[0] for i in sim_scores]
    hotels = base.iloc[hotel_indices]
    hotels = hotels.sort_values(by = "price_per_night")

    #menambahkan informasi hotel yang menjadi input sebagai perbandingan
    hotels = hotels.append(base[base['name'] == hotel])

    #dengan menggunakan iloc, kita bisa panggil balik berdasarkan index dari hotel indices
    return hotels

content_recommender("Hotel Tentrem Yogyakarta")
```

Out[37]:

	name	id	description	province	city	address	price_per_night
378	Hotel Asri Graha Yogyakarta	h1197	pilihan akomodasi ideal pasangan backpacker li...	daerah istimewa yogyakarta	jogja	jl veteran 184 umbulharjo yogyakarta yogyakart...	262500
20	Aloha Hotel	h1166	penginapan tenang nyaman yogyakarta aloha hote...	daerah istimewa yogyakarta	jogja	jl prawirotaman mg iii 573 b prawirotaman yogy...	295000
1041	Wisma Ary's Hotel	h1115	pilihan akomodasi ideal backpacker liburan kel...	daerah istimewa yogyakarta	jogja	jl suryodiningratan 29 prawirotaman mantrijero...	300000
953	The Cube Hotel	h0999	sebuah hotel nyaman lengkap terjangkau yogyaka...	daerah istimewa yogyakarta	jogja	jl parang tritis 16 yogyakarta indonesia	432500
115	Bueno Colombo Hotel	h1094	penginapan cocok bisnis bersantai terletak str...	daerah istimewa yogyakarta	jogja	jl raya yogya solo km 14 kalasan yogyakarta in...	536250
543	Hotel Tentrem Yogyakarta	h1112	penginapan cocok bisnis bersantai terletak str...	daerah istimewa yogyakarta	jogja	jl p mangkubumi 72a jetis yogyakarta indonesia	1492330

Terlihat bahwa dari output rekomendasi hotel yang dihasilkan semua lokasinya berada di provinsi dan kota yogyakarta, biarpun saat memanggil fungsi kita tidak menentukan lokasi yang diinginkan. Hal ini karena 'bag of words' yang digunakan telah memasukkan fitur lokasi sehingga model akan merekomendasikan hotel dari lokasi yang sama karena memiliki kemiripan yang lebih tinggi. Selain itu, terdapat juga hotel yang memiliki hampir sama persis dengan Hotel Tentrem, yaitu Bueno Colombo Hotel

6.3 Kasus 3 : Kasus lain

Di bagian ini kita ambil contoh kasus hotel "La Oma", masih menggunakan model content based system seperti pada kasus nomor 2

In [38]:

```
content_recommender("La Oma")
```

Out[38]:

	name	id	description	province	city	address	price_per_night
370	Hotel Alpha Classica	h0834	alamat jalan tangkuban perahu 25 lembang waktu...	jawa barat	bandung	jalan tangkuban perahu 25 lembang	300000
169	D'Talent Hotel	h1038	alamat jalan prawirotaman 3 66 waktu check sta...	daerah istimewa yogyakarta	jogja	jalan prawirotaman 3 66	300000
550	Hotel Tulips	h1050	alamat jalan tirtodipuran 42 waktu check stand...	daerah istimewa yogyakarta	jogja	jalan tirtodipuran 42	411670
461	Hotel Kirana	h1046	alamat jl prawirotaman 45 waktu check standar ...	daerah istimewa yogyakarta	jogja	jl prawirotaman 45	427500
206	Family Budget Hotels F77	h1044	alamat jalan r e martadinata 73 waktu check st...	daerah istimewa yogyakarta	jogja	jalan r e martadinata 73	650000
621	La Oma	h0840	alamat jalan cijeruk 62 lembang waktu check st...	jawa barat	bandung	jalan cijeruk 62 lembang	800000

Terlihat bahwa di kasus ini hanya satu hotel hasil rekomendasi yang memiliki lokasi yang sama dengan hotel La Oma, dan sisanya berasal dari kota jogja. Jika kita lihat pada kolom description, kesemuanya memiliki description yang hampir sama persis, sehingga model akan menilai bahwa hotel tersebut mirip satu sama lain

In [39]:

```
# menampilkan perbandingan deskripsi hotel La Oma dengan Hotel D'Talent
print(base.iloc[621,2])
print("\n",base.iloc[169,2])
```

```
alamat jalan cijeruk 62 lembang waktu check standar 14 00 waktu check plan mempunyai prioritas lebih
besar waktu check standar 12 00 waktu check plan mempunyai prioritas lebih besar jumlah kamar 0 sing
le 0 double 0 twin 0 suite 0 lainnya 0 tahun dibangun tahun renovasi

 alamat jalan prawirotaman 3 66 waktu check standar 14 00 waktu check plan mempunyai prioritas lebih
besar waktu check standar 12 00 waktu check plan mempunyai prioritas lebih besar jumlah kamar 0 sing
le 0 double 0 twin 0 suite 0 lainnya 0 tahun dibangun tahun renovasi
```

7. Evaluasi Keakuratan Model

Keakuratan model untuk kasus ini dapat dievaluasi menggunakan metric ***Precision*** dan ***Recall***. Karena content based system berdasarkan pada kemiripan antara rekomendasi dan acuan/input, kita dapat melihat berapa banyak hasil rekomendasi yang benar2 relevan (True Positive) dan melakukan perhitungan dari 2 metrics tersebut. Formula yang dapat digunakan adalah :

$$\text{recommender system precision:} \qquad P = \frac{\text{\# of our recommendations that are relevant}}{\text{\# of items we recommended}}$$

$$\text{recommender system recall:} \qquad r = \frac{\text{\# of our recommendations that are relevant}}{\text{\# of all the possible relevant items}}$$

Sehingga pada contoh kasus 3, karena secara umum hanya terdapat 1 hasil rekomendasi yang relevan (hotel yang berada di Bandung), maka presisinya adalah P = 1/5 = 20%

8. Cara Meningkatkan Sistem Rekomendasi

1. Untuk content based recommender : menambahkan filter lokasi maupun range harga untuk mendapatkan rekomendasi yang lebih akurat
2. Menambah fitur seperti jenis fasilitas yang tersedia di setiap hotel
3. Menggabungkan kedua model (simple recommender dan content based) dan menghasilkan rekomendasi hotel yang mirip namun rating/popularitas yang lebih tinggi

Referensi

Building Recommender System using Python :(https://academy.dqlab.id/main/package/project/212 (https://academy.dqlab.id/main/package/project/212))

Building Recommender System using Similarity Function in Python: (https://academy.dqlab.id/main/package/project/214 (https://academy.dqlab.id/main/package/project/214))

https://towardsdatascience.com/building-a-content-based-recommender-system-for-hotels-in-seattle-d724f0a32070?gi=13c7fbb84447 (https://towardsdatascience.com/building-a-content-based-recommender-system-for-hotels-in-seattle-d724f0a32070?gi=13c7fbb84447)

https://medium.com/@ksnugroho/dasar-text-preprocessing-dengan-python-a4fa52608ffe (https://medium.com/@ksnugroho/dasar-text-preprocessing-dengan-python-a4fa52608ffe)

https://medium.com/data-folks-indonesia/recommendation-system-dengan-python-content-based-filtering-part-2-222a8c365add (https://medium.com/data-folks-indonesia/recommendation-system-dengan-python-content-based-filtering-part-2-222a8c365add)

https://towardsdatascience.com/recommendation-systems-models-and-evaluation-84944a84fb8e (https://towardsdatascience.com/recommendation-systems-models-and-evaluation-84944a84fb8e)

Banik,Rounak (2018) *Hands-On Recommendation Systems with Python - Start building powerful and personalized, recommendation engines with Python*. Birmingham:Packt Publishing Ltd