

# Data Mining Review Hotel The Apurva Kempinski Bali Menggunakan BeautifulSoup

Dibuat Oleh: Stefannus Christian

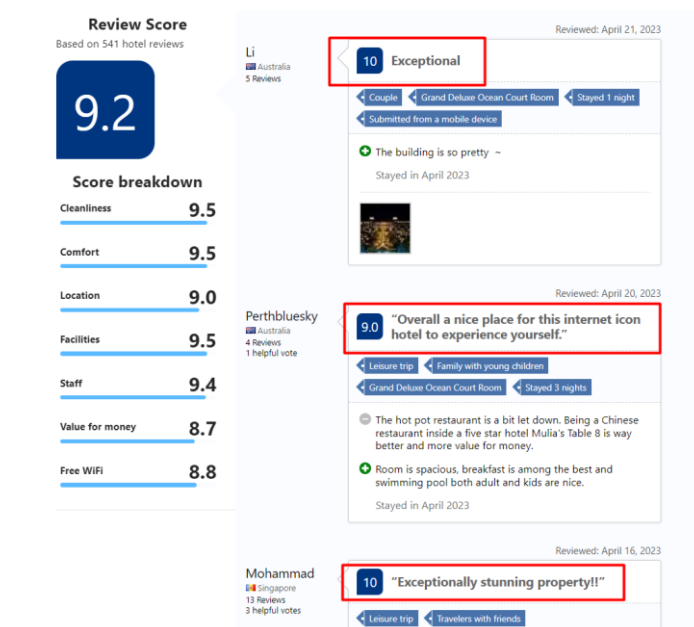
Link Github Project: <https://github.com/StefannusChristian/Data-Mining/tree/main/apurva-kempinski-bali-web-scraping>

Berikut merupakan link – link referensi belajar yang saya gunakan ketika mengerjakan proyek ini:

1. Link Belajar Beautiful Soup: <https://www.youtube.com/watch?v=XVv6mJpFOb0>
2. Link Belajar WordCloud: <https://www.youtube.com/watch?v=HcKUU5nNmrs>

Disini saya menganalisis review penjunjung hotel yang termuat dalam website booking.com dengan link berikut ini: [https://www.booking.com/reviews/id/hotel/the-apurva-kempinski-bali.html?sid=d5b706fb1e392d7bdb21e2beb7949c62&aid=356980&order=featuredreviews&label=gog235jc-1FCA0oaEIZdGhLLWFwdXJ2YS1rZW1waW5za2ktYmFsaUgzWANOalgBAZgBMbgBF8gBDNgBAegBAfgBAogCAagCA7gCnNKLogbAAgHSAiRhNTZlZDcxNS0yNTlzMTRmY2QtOTlwMy05MmRhYTYVIYzdkOWLYAgXgAgE&hp\\_nav=0&page=1&customer\\_type=total&rows=75&old\\_page=0&r\\_lang=en](https://www.booking.com/reviews/id/hotel/the-apurva-kempinski-bali.html?sid=d5b706fb1e392d7bdb21e2beb7949c62&aid=356980&order=featuredreviews&label=gog235jc-1FCA0oaEIZdGhLLWFwdXJ2YS1rZW1waW5za2ktYmFsaUgzWANOalgBAZgBMbgBF8gBDNgBAegBAfgBAogCAagCA7gCnNKLogbAAgHSAiRhNTZlZDcxNS0yNTlzMTRmY2QtOTlwMy05MmRhYTYVIYzdkOWLYAgXgAgE&hp_nav=0&page=1&customer_type=total&rows=75&old_page=0&r_lang=en)

Review – review tulisan di kotak merah pada gambar dibawah ini adalah review yang akan saya analisis.



Langkah – Langkah:

1. Install Library Python yang dibutuhkan

Jalankan ketiga perintah dibawah ini pada untuk menginstall library beautifulsoup yang akan digunakan untuk scraping, menginstall library wordcloud yang akan digunakan

untuk membuat wordcloud dari hasil scraping, dan menginstall library lxml yang dibutuhkan untuk memproses html.

- *pip install beautifulsoup4*
- *pip install wordcloud*
- *pip install lxml*

## 2. Import Library yang diperlukan

```
from requests import get
from re import sub
from bs4 import BeautifulSoup
from wordcloud import WordCloud, STOPWORDS
from time import perf_counter
```

- Library requests method get digunakan untuk mengambil text dari situs booking.com.
- Library re (regex) method sub digunakan untuk membersihkan text hasil dari scraping seperti membuang spasi berlebih pada text hasil scraping.
- Library BeautifulSoup akan digunakan untuk web scraping.
- Library WordCloud akan digunakan untuk membuat wordcloud.
- Library time method perf\_counter akan digunakan untuk menghitung waktu eksekusi program seperti waktu eksekusi scraping per page dan juga waktu eksekusi membuat word cloud.

## 3. Scraping Web Booking.com

```
27 def scrape_reviews():
28     scrape_start_time = perf_counter()
29     all_reviews = []
30     for page in range(1,9):
31         page_start_time = perf_counter()
32         print(f"Scraping page {page} ...")
33         url = f"https://www.booking.com/reviews/id/hotel/the-apurva-kempinski-bali.html?aid=356980&
label=gog235jc-10CA9oaEIZdGhlLWFwdXJ2YS1rZWlwaW5za2ktYmFsaUgzWAAoaIqBAZg8MBgBF8gBDNgBA-gBAfgBAogCAagCA7gCnNKLogbAAgHSAiRhNTZlZDcxNS0yNTIz
TRmY2QtOTlWMy05MmRhYTlVYzdkOWLYAgTgAgE&sid=d5b706fb1e392d7bdb21e2beb7949c62&customer_type=total&hp_nav=0&old_page=0&order=featuredreviews&
page={page}&r_lang=en&rows=75&"
34
35         html_text = get(url).text
36         soup = BeautifulSoup(html_text, 'lxml')
37         reviews = soup.find_all('div', class="review_item_header_content")
38
39         for word in reviews:
40             try:
41                 review = word.span.text.strip()
42                 review = remove_special_characters(review)
43                 review = correct_typo(review)
44                 all_reviews.append(review.strip().lower())
45             except: pass
46
47         page_end_time = perf_counter()
48
49         print(f"Scraping page {page} finished!")
50         print(f"Scraping page {page} took {page_end_time-page_start_time:.4f} seconds\n")
51
52     scrape_end_time = perf_counter()
53     print(f"Scraping took {scrape_end_time-scrape_start_time:.4f} seconds!\n")
54
55     return all_reviews
```

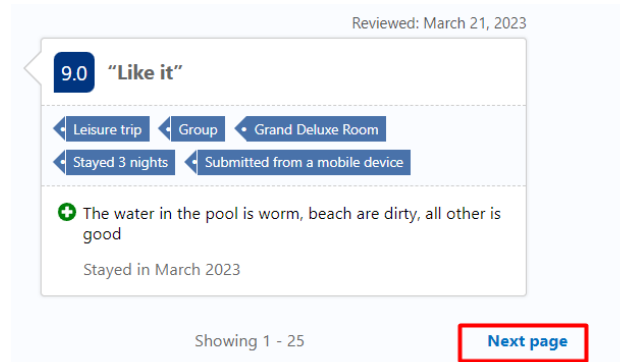
Disini saya membuat function scrape\_reviews() untuk scraping web booking.com untuk hotel Apurva Kempinski Bali. Saya akan menjelaskan kode saya line per line dengan memberikan terlebih dahulu cuplikan kode kemudian menjelaskan apa yang kode tersebut lakukan.

```

28     ... scrape_start_time = perf_counter()
29     ... all_reviews = []

```

Variabel `scrape_start_time` akan mencatat waktu Ketika function `scrape_review` dijalankan. Variabel ini nantinya akan digunakan untuk menghitung waktu yang digunakan untuk menjalankan scraping. Variabel `all_reviews` digunakan untuk menyimpan seluruh review – review hasil scraping.



Terdapat 8 page review pada website ini. Website berikutnya dapat diakses dengan menekan next page seperti gambar diatas. Tetapi disini, saya menemukan bahwa URL yang digunakan untuk page berikutnya dapat diubah dari bagian url. Dapat dilihat dari gambar dibawah ini bahwa saya tinggal mengganti url web pada bagian `{page}` dan reviewnya pun akan berubah ke review halaman `{page}`.

```

url = f"https://www.booking.com/reviews/id/hotel/the-apurva-kempinski-bali.html?aid=356980&
label=gog235jc-1DCA00aEIZd6hLLWFwdXJ2YS1rZW1waW5za2ktYmFsaUgzWANoaIgBAZgBMbgBF8gBDNgBA-gBAfgBAogCAagCA
7gCnNKLogbAAgHSAiRhNTZLZDcxNS0yNTIzLTRmY2QtOTIwMy05MmRhYTVLYzdkOWLYAgTgAgE&
sid=d5b706fb1e392d7bdb21e2beb7949c62&customer_type=total&hp_nav=0&old_page=0&order=featuredreviews&
page={page}&r_lang=en&rows=75&"

```

```

for page in range(1,9):
    page_start_time = perf_counter()
    print(f"Scraping page {page} ...")
    url = f"https://www.booking.com/reviews/id/hotel/the-apurva-kempinski-bali.html?aid=356980&
label=gog235jc-1DCA00aEIZd6hLLWFwdXJ2YS1rZW1waW5za2ktYmFsaUgzWANoaIgBAZgBMbgBF8gBDNgBA-gBAfgBAogCAagCA
7gCnNKLogbAAgHSAiRhNTZLZDcxNS0yNTIzLTRmY2QtOTIwMy05MmRhYTVLYzdkOWLYAgTgAgE&
sid=d5b706fb1e392d7bdb21e2beb7949c62&customer_type=total&hp_nav=0&old_page=0&order=featuredreviews&
page={page}&r_lang=en&rows=75&"

```

Maka dari itu karena adanya 8 page pada website ini, saya melakukan for loop untuk mengiterasi dari page 1 – 8. Variabel `page_start_time` digunakan untuk menghitung start waktu dari ketika halaman `{page}` mulai di scrape dan nantinya akan digunakan untuk menghitung waktu yang dibutuhkan untuk scraping halaman `{page}`.

```

html_text = get(url).text
soup = BeautifulSoup(html_text, 'lxml')

```

Disini variable `html_text` digunakan untuk mendapatkan struktur html dari url yang diberikan. Variabel `soup` digunakan untuk menginisialisasi BeautifulSoup yang akan

digunakan untuk scraping. Saya tidak akan menjelaskan secara detail apa yang dilakukan kedua variabel diatas, penjelasan lebih lanjut dapat ditonton [disini](#).

Cara kerja beautifulsoup (bs) adalah bs dengan bantuan lxml (parser) akan mencari tag atau class dari elemen yang ingin di scrape. Disini, menggunakan dev tools browser untuk inspect element yang ingin dicari apa nama classnya dan class tersebut berada pada tag apa.

```
▼ <div class="review_item_header_content
    high_score_word
    review_item_header_scoreword
"> == $0
    <span itemprop="name"> Exceptional </span>
</div>
```

Dapat dilihat bahwa kata review yang ingin saya ambil yaitu Exceptional berada pada di tag span dengan parent tag div dengan class review\_item\_header\_content. Maka cara saya mendapatkan seluruh review adalah dengan mencari seluruh div tag dengan class review\_item\_header\_content. Berikut kode untuk melakukan hal ini.

```
reviews = soup.find_all('div', class_="review_item_header_content")
```

Tetapi disini variabel reviews belum secara spesifik mendapatkan kata review yang diinginkan. Reviews disini bentuknya adalah list yang menyimpan seluruh div dengan class review\_item\_header\_content.

```
for word in reviews:
    try:
        review = word.span.text.strip()
        review = remove_special_characters(review)
        review = correct_typo(review)
        all_reviews.append(review.strip().lower())
    except: pass
```

Berikut cara untuk mendapatkan kata review yang diinginkan. Iterasilah list reviews dan berdasarkan struktur kode html yang telah saya berikan diatas, review word nya terdapat pada span sehingga kita perlu mengakses span tersebut dengan cara word.span. Kemudian .text digunakan untuk mendapatkan kata dari span tersebut dan .strip() digunakan untuk menghilangkan spasi berlebih dari string tersebut. Kemudian review tersebut akan dibersihkan dan di append ke list all\_reviews. Saya menggunakan function lower() untuk membuat review menjadi huruf kecil.

```
def remove_special_characters(review: str):
    review = review.replace("😊", "")
    review = review.replace("👍", "")
    review = review.replace("'", "")
    review = review.replace("!", "")
    review = review.replace(".", "")
    review = review.replace(")", "")
    review = review.replace("(", "")
    review = review.replace(":", "")
    review = review.replace("-", "")
    review = review.replace(" ", "")
    return review
```

```
def correct_typo(review: str):
    review = review.replace("pleasent", "pleasant")
    review = review.replace("recomended", "recommended")
    review = review.replace("recommend", "recommended")
    review = review.replace("exceptionally", "exceptional")
    return review
```

Kedua function diatas saya apply untuk membersihkan review sebelum dimasukkan kedalam list all\_reviews. Saya replace tanda baca seperti titik, koma, dll. Saya juga membenarkan typo yang saya temukan di review tersebut seperti pleasant yang seharusnya pleasant.

```
... page_end_time = perf_counter()
... print(f"Scraping page {page} finished!")
... print(f"Scraping page {page} took {page_end_time-page_start_time:.4f} seconds\n")
...
    scrape_end_time = perf_counter()
    print(f"Scraping took {scrape_end_time-scrape_start_time:.4f} seconds!\n")
...
    return all_reviews
```

Kemudian variabel page\_end\_time digunakan sebagai tanda bahwa scraping page tersebut sudah selesai. Kemudian cara untuk menghitung berapa waktu yang digunakan untuk scraping satu page adalah dengan mengurangi waktu akhir dengan waktu awal. Begitu juga cara untuk menghitung waktu total yang digunakan untuk scraping seluruh page. Function ini kemudian akan return all\_reviews yang isinya adalah review words yang sudah di bersihkan.

#### 4. Menyimpan hasil scraping ke file txt

```
57 def clean_reviews(reviews: list): return list(filter(None, reviews))
58
59 v def convert_review_to_paragraph(reviews: list):
60     review_text = ". ".join(reviews)
61     review_text = sub(" ", " ", review_text)
62     return review_text
63
64 v def save_review_to_txt_file(path: str, filename: str):
65     all_reviews = clean_reviews(scrape_reviews())
66     print("Scraping Apurva Kempinski Bali Reviews From Booking.com Website Finished!")
67     review_text = convert_review_to_paragraph(all_reviews)
68     with open(path+filename, "w") as text_file: text_file.write(review_text)
```

Untuk menconvert list all\_reviews menjadi word cloud, list all\_reviews perlu dibuat menjadi file txt terlebih dahulu. Yang saya lakukan pertama adalah membersihkan lagi list all\_reviews dengan menghilangkan NaN values, kemudian saya menggunakan method join untuk menggabungkan isi dari list tersebut dan membuatnya menjadi satu string Panjang (paragraph). Kemudian saya menggunakan method regex yaitu sub untuk menghilangkan spasi spasi dobel yang

ada pada paragraph tersebut. Setelah itu saya convert paragraph tersebut ke file txt. Hasil dari file txt nya adalah sebagai berikut.

```
scraping_word_cloud.py M review.txt
apurva-kempinski-bali-web-scraping > review.txt

1 exceptional overall a nice place for this internet icon hotel to experience yourself exceptionally stunning property incredible place to stay dream hotel you should
stay at least once it was excellent overall wonderful beautiful hotel best hotel ive ever had the pleasure of staying in enjoyed the lagoon very much and the staff
were very friendly unimaginable resort relaxing and enjoyable stay exceptional wonderful exceptional one of the best resorts we have every stayed and cant wait to
come back exceptional exceptional cant wait to go back magnificent and beautiful worth every penny massive hotel lovely the stay was great super friendly staff and i
already returned one more time to kempinski like it new and tidy warm welcome and friendly hotel will come back luxurious familyfriendly hotel most memorable stay i
have ever had exceptional this hotel is the highlight of my bali trip excellent property but needs to improve their breakfast and its quite dark when you walk in
halls towards the room exceptional great for families amazing swimming pools exceptional best hotel in nusa dua very recommended overall this hotel is highly
recommended real luxury extremely comfortable rooms and bed great food and friendly staff great place to take a whole vacation in one place wonderful awesome and
definitely will return & recommended this hotel the hotel is a perfect place for family trip to spend and to enjoy the property and its facility perfect stay amazing
exceptional fantastic experience its the best resort i have ever stayed at so far exceptional wonderful excellent place to chill with good food and good wine
definitely will come back again amazing the villa was very comfortable and relaxing everything is so amazing will definitely come back again later despite the little
hiccups we had a really good time good but can be improved overall that was a wonderful stay we will definitely go back amazing decor beautiful location and culinary
variety that was unique exceptional ive had one of the most memorable hotel stay thanks to the experience i had at apurva i look forward to going again possibly the
best hotel on the planet every aspect of our stay from the check in to the check out was spectacular and a wonderful very good very good would come back for sure
very good experience overall exceptional very luxurious pools are great although ive seen better pools at cheaper hotels in bali immaculate hotel immaculate
facilities and lovely location impressive location restaurant koral alone is worth the experience the best resort in bali most excellence hotel that we had stay in
bali island great resort with luxurious 5 stars services we had a pleasant and comfortable stay pleasant good fantastic family holiday trip bad good good bad
experience very poor need to improve loved the building and facilities overall it was a good stay besides the long check in time the view and deco at the hotel is
superb very poor good amazing vacation and absolutely worth every penny very good excellent stay very hospitable excellent service and facilities enjoyed our stay
there wonderful wonderful very good very good very comfortable and memorable stay wonderful amazing i think the hotel is understaffed during this peak season so
while the facilities are great the service was a bit disa exceptional exceptional wonderful exceptional exceptional exceptional exceptional exceptional wonderful
exceptional exceptional exceptional exceptional very good exceptional exceptional exceptional exceptional exceptional wonderful exceptional wonderful exceptional
exceptional exceptional exceptional exceptional very good exceptional exceptional exceptional exceptional exceptional wonderful exceptional wonderful exceptional truly
exceptional my best hotel experience ever wonderful exceptional very good the whole hotel complex was amazing and service was outstanding thank you very much to the
kempinski team for making our exceptional exceptional wonderful wonderful wonderful wonderful wonderful wonderful wonderful wonderful wonderful wonderful wonderful
exceptional wonderful wonderful wonderful wonderful wonderful exceptional exceptional good good good fair fair good exceptional wonderful exceptional wonderful very good
wonderful wonderful exceptional exceptional wonderful
```

File txt inilah yang akan digunakan untuk membuat wordcloud.

## 5. Membuat wordcloud

```
def convert_review_to_wordcloud(reviews: str, wordcloud_file_name: str):
    print("Converting Reviews To WordCloud ...")
    wordcloud_start_time = perf_counter()
    sw = list(STOPWORDS)
    stopwords = list(set(["a", "overall", "place", "for", "this", "internet", "icon", "hotel", "to", "yourself", "property", "place", "stay", "the", "you", "should", "at", "least",
    "once", "it", "the", "very", "much", "and", "one", "of", "were", "really", "walk", "need", "wait", "team", "think", "spend", "bit", "building", "hiccups", "ou", "deco", "dua", "peak",
    "complex", "decor", "koral", "every", "towards", "apurva", "extremely", "sure", "nusa", "disa", "little", "planet", "lagoon", "cant", "go", "take", "aspect", "halls", "although",
    "needs", "seen", "returned", "kempinski", "ive", "penny", "look", "trip", "besides", "later", "already", "long", "back", "alone", "return", "staying", "come", "will", "bali",
    "everything", "swimming", "room", "stayed", "bed", "truly", "possibly", "welcome", "dark", "making", "improved", "real", "thank", "better", "staff", "island", "whole", "season",
    "going", "forward", "pools", "rooms", "check", "time", "despite", "hotels", "highly", "improve", "definitely"
    ]))

    sw.extend(stopwords)
    sw = list(set(sw))

    wc = WordCloud (
        background_color='white',
        stopwords=sw,
    )

    wc.generate(reviews)
    wc.to_file(wordcloud_file_name+'.png')

    wordcloud_end_time = perf_counter()
    print("\nConverting Reviews To WordCloud Finished!")
    print(f"Converting To WordCloud Takes {wordcloud_end_time - wordcloud_start_time:.4f} seconds\n")
```

Saya menggunakan library wordcloud pada python untuk membuat wordcloud.

Untuk referensi nya saya belajar [disini](#). Stopwords digunakan untuk memfilter kata – kata apa saja yang tidak diinginkan pada wordcloud. Ketika menginstall wordcloud, default STOPWORDS otomatis akan terinstall juga tetapi default STOPWORDS yang disediakan ini tidak cukup sehingga saya menambahkan stopwords berikut ini secara manual agar kata – kata ini tidak ditampilkan pada wordcloud.

```
stopwords = list(set(["a", "overall", "place", "for", "this", "internet", "icon", "hotel",
    "to", "yourself", "property", "place", "stay", "the", "you", "should", "at", "least", "once",
    "it", "the", "very", "much", "and", "one", "of", "were", "really", "walk", "need", "wait",
    "team", "think", "spend", "bit", "building", "hiccups", "ou", "deco", "dua", "peak", "complex",
    "decor", "koral", "every", "towards", "apurva", "extremely", "sure", "nusa", "disa", "little",
    "planet", "lagoon", "cant", "go", "take", "aspect", "halls", "although", "needs", "seen",
    "returned", "kempinski", "ive", "penny", "look", "trip", "besides", "later", "already",
    "long", "back", "alone", "return", "staying", "come", "will", "bali", "everything",
    "swimming", "room", "stayed", "bed", "truly", "possibly", "welcome", "dark", "making",
    "improved", "real", "thank", "better", "staff", "island", "whole", "season", "going",
    "forward", "pools", "rooms", "check", "time", "despite", "hotels", "highly", "improve",
    "definitely"
    ]))
```



Output program di console ketika di run adalah sebagai berikut.

```
Scraping Apurva Kempinski Bali Reviews From booking.com Website Starting ...

Scraping page 1 ...
Scraping page 1 finished!
Scraping page 1 took 1.3910 seconds

Scraping page 2 ...
Scraping page 2 finished!
Scraping page 6 ...
Scraping page 6 finished!
Scraping page 6 took 1.1317 seconds

Scraping page 7 ...
Scraping page 7 finished!
Scraping page 7 took 1.1887 seconds

Scraping page 8 ...
Scraping page 8 finished!
Scraping page 8 took 1.5663 seconds

Scraping took 11.1786 seconds!

Scraping Apurva Kempinski Bali Reviews From Booking.com Website Finished!
Converting Reviews To WordCloud ...

Converting Reviews To WordCloud Finished!
Converting To WordCloud Takes 0.4934 seconds

Program Takes 11.6755 seconds to execute
```

Hasil WordCloud

