Jogjakarta Cafe Hopping:

Google Maps Data Scraping with Python & Google Collab

During pandemic time, I've always thinking about how to get a proper vacation, yes you heard it right, vacation. While the government are continuously promoting #dirumahaja (stay at home) campaign, most of the province in Indonesia are right now trying to create a travel bubble that open up the chance for domestic traveler to go to the designated province.

As for myself, I'm really hyped for this and planning to go with 2 of my bestfriends for a roadtrip from Jakarta to Jogjakarta for 3 days. Unfortunately, because of the small amount of time that I propose, I might not get a chance to explore Jogjakarta thoroughly. So then me and my bestfriends figured out what if we just focus our vacation on Cafe Hopping around our hotel in Jogjakarta, because there are lots of Cafes that is really aesthetic and have the best foods and drinks. But here is come the problem, we don't know which Cafe is good around our Hotel. We can definitely Google it, but it will take a long time to just compile it one by one.

Business Question:

1. Can we have a compilation of Cafe near our hotel that consist of the name of the cafe, the address, the ratings, and the latitude and longitude so it will be easier to get direction?
2. Can we have a map that shows the cafe around our hotel?

Methods

1. Google Maps API to gain and scrape location data
2. Pandas Library to export dataframe
3. Kepler.gl to visualize the data
4. I also use Google Colab to run the script

Step One: From which area do you like to start your Cafe Hopping?

For me, I would like to first define where I'm going to stay, then from there I will define the distance radius of my Cafe Hopping. Since I'm going to stay at Artotel Jogjakarta that is located in Kaliurang, I'm going to use Artotel Coordinates as a center point and define 1.5 KM in radius for my Cafe Hopping adventure. Therefore, the parameters would be "Cafe", "Artotel" (in coordinate), and "1.5 KM". Translated into Python, it would be:

In [20]:
```
coordinates = [-7.756564775789093, 110.38207221506629]
keywords = ['cafe']
radius = 1500
api_key = 'AIzaSyAmEGzNexQM5reOw-9MHPjbtlabnZaPMKXHk'
```

By adding a keywords, it will let us get any places that are registered as cafe or whose name has 'cafe' on it instead of using 'type' or 'name' where it will return on specific parameters.

For example, if we use 'type', it will return only place that are belongs to 'cafe type'. Since we use 'keywords', it will return both the 'type' and 'name' that contains cafe, so then Starbucks and Cafe Gelato will both being query.

Step Two: Import Libraries Needed

In [7]:
```
import pandas as pd
import numpy as np
import requests
import json
import time
from google.colab import files
```

Yes, since I'm using Google Colab, I also import files library from google colab to open and save data.

Step Three: Craft The Code

In [ ]:
```
for coordinate in coordinates:
    for keyword in keywords:

        url = 'https://maps.googleapis.com/maps/api/place/nearbysearch/json?location='+coo

        while True:
            print(url)
            response = requests.get(url)
            jj = json.loads(response.text)
            results = jj['results']
            for result in results:
                name = result['name']
                place_id = result['place_id']
                lat = result['geometry']['location']['lat']
                lng = result['geometry']['location']['lng']
                rating = result['rating']
                types = result['types']
                vicinity = result['vicinity']

                data = [name, place_id, lat, lng, rating, types, vicinity]

                final_data = []

                final_data.append(data)

            time.sleep(5)

            if 'next_page_token' not in jj:
                break
            else:
                next_page_token = jj['next_page_token']
                url = 'https://maps.googleapis.com/maps/api/place/nearbysearch/json?key='+str

labels = ['Place Name', 'Place ID', 'Latitude', 'Longitude', 'Ratings', 'Types', 'Vici
```

The code above provide us to get the place's name, id, latitude-longitude, rating, type and vicinity for every coordinates and keyword. Since Google only shows 20 data points in each page, we have to insert 'next_page_token' to scrape the next page's data. Let's say there are 30 cafes around Artotel Jogjakarta, then Google will show the data in two pages. If there are 50, then it would show in 3 pages.

The maximum data point that we can extract are only 60 datas. It's a regulation from Google. For example, if there are 200 cafes around Artotel Jogjakarta in the radius of 1.5KM, only 60 out of 200 cafes that would be generated. In order to prevent bias, make sure to control our radius accordingly. Do not make the radius too wide or even to small. Both would not be efficient, hence understanding the context first would be a better option.

Step Four: Save The Generated Data to Local Machine

In [ ]:
```
export_dataframe_1_medium = pd.DataFrame.from_records(final_data, columns = labels)
export_dataframe_1_medium.to_csv('export_dataframe_1_medium.csv')
```

Step Five: Knit The Code

In [13]:
```
import pandas as pd
import numpy as np
import requests
import json
import time
from google.colab import files
final_data = []

#Parameters
coordinates = [-7.756564775789093, 110.38207221506629]
keywords = ['cafe']
radius = 1500
api_key = 'AIzaSyAmEGzNexQM5reOw-9MHPjbtlabnZaPMKXHk'

for coordinate in coordinates:
    for keyword in keywords:

        url = 'https://maps.googleapis.com/maps/api/place/nearbysearch/json?location='+coo

        while True:
            print(url)
            response = requests.get(url)
            jj = json.loads(response.text)
            results = jj['results']
            for result in results:
                name = result['name']
                place_id = result['place_id']
                lat = result['geometry']['location']['lat']
                lng = result['geometry']['location']['lng']
                rating = result['rating']
                types = result['types']
                vicinity = result['vicinity']

                data = [name, place_id, lat, lng, rating, types, vicinity]

                final_data.append(data)

            time.sleep(5)

            if 'next_page_token' not in jj:
                break
            else:
                next_page_token = jj['next_page_token']
                url = 'https://maps.googleapis.com/maps/api/place/nearbysearch/json?key='+str

labels = ['Place Name', 'Place ID', 'Latitude', 'Longitude', 'Ratings', 'Types', 'Vic
export_dataframe_1_medium = pd.DataFrame.from_records(final_data, columns = labels)
export_dataframe_1_medium.to_csv('export_dataframe_1_medium.csv')
```

```
https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=-7.756564775789
093, 110.38207221506629&radius=1500&keyword=cafe&key=AIzaSyAmEGzNexQM5reOw-9MHPjbtlabnZaP
MKXHk
https://maps.googleapis.com/maps/api/place/nearbysearch/json?key=AIzaSyAmEGzNexQM5reOw
-9MHPjbtlabnZaPMKXHk&pagetoken=ATYBwG3HnZEJvPpIGQ2oeqa2PEiDFQ2Xvs6z5oUlsKhjmDNyndxvL_W6
PRmX1E4BUgDmCAs8eAbAZnCFvTdp7RzonDHXUHMDlmqGR_lvpe4uaxXBizhZA_4kyhXDVHEaa5kaaWmv-
jfyJikzkAkeuBVRGNTTEIDn-BOWqhCGalhqD_3WqS_qKmscTtgzyqMpFJq6_9iaOwlfiI-9XDbVvqxEE9qhaXD
TF1vmcN6Qp3rPmVaVqgk-cMiPwjDnRzgTATvvGqsRJuraJu2UJbSaDmRIi-WrSuFZHjMyyzpWVRV3YQ3RAlu
mbcUAp3eJuajpPuTwwGanYAABGmmn7T2HL-nt-7sk1PFeOlitaNhCiilDoPMhoMG7Q2j4xe4IXhSrzXOVQBm
v-IB4ZmbZwPwwagXpzAmf64T1YOhLIcXa_Pt-ygsxlNNlSNyzwdxcLk3ZcCTvKTFEtcqBmcSAAMbm
NtcX8GVEaJWVbd894DhgEKXjcueu_tEqnGeIO4THWrnbrO8bW4bzPlfgTWlVwpFTAmGxSWhmkKWmQ8Ls_6S
eMBgMGZFlo_-TrUOoaYfhWwC-uDG2aZLfs1TLNVcnrPnapAWFK3QFdL_ZtWQfdUhJvNg-A5CfmwQ
```

In [ ]:

Step Six: Download the Dataset

Download your data from the Google Colab files. Click the folder icon on the left-pane, then download your data.

In [ ]:
```
from IPython.display import Image
from IPython.core.display import HTML
Image(url= "https://github.com/fadlanawirya/Cafe-Hopping-in-Jogjakarta/blob/main/Asse
```

Out[3]:

Step Seven: Share The Data

Now you already downloaded the data in CSV format. In order to better understand the data, you can visualized the data using any tools you are familiar with, R, Python, Tableau, Kepler.gl, etc. In my case, I use Kepler.gl to visualize the data. Kepler.gl is a data agnostic, WebGL empowered, high-performance web application for geospatial analytic visualizations.

In [4]:
```
from IPython.display import Image
from IPython.core.display import HTML
Image(url= "https://github.com/fadlanawirya/Cafe-Hopping-in-Jogjakarta/blob/main/Asse
```

Out[4]:

This is how it looks like in Kepler.gl map:
(p.s. I edit it a bit in Adobe Illustrator)

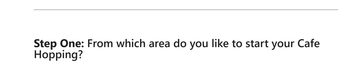
In [7]:
```
Image(url= "https://github.com/fadlanawirya/Cafe-Hopping-in-Jogjakarta/blob/main/Asse
```

Out[7]:

And to see the point based on ratings of each cafes...
(the size of the embedded image is purposely big, so mind to scroll)

In [9]:
```
Image(url= "https://github.com/fadlanawirya/Cafe-Hopping-in-Jogjakarta/blob/main/Asse
```

Out[9]:



That's 52 cafes around Artotel Hotel Jogjakarta, you can add their name also if you want. Within this, I'm good to go for Cafe Hopping adventure in Jogjakarta with my bestfriends!

Big thanks to Regita H.Zokia for her Medium article that become my based of learning and therefore created this case study!