Praktikum Mengumpulkan Data

Tahap 1: Menentukan Kebutuhan Data

Langkah-langkah praktikum dalam menentukan kebutuhan data

- Identifikasi kebutuhan data berdasarkan tujuan teknis data science dengan cara:
    - Buat daftar fitur-fitur data yang akan digunakan dalam proyek, dan jelaskan alasan pemilihan setiap fitur tersebut.

      Contoh 1: Anda bekerja pada proyek untuk mengidentifikasi pelanggan loyal di sebuah bank. Untuk melakukan ini, Anda membutuhkan data seperti umur, pendapatan, jumlah transaksi, dan skor kredit dengan deskripsi sebagai berikut:

        - *umur*: untuk melihat distribusi usia nasabah.
        - *pendapatan*: untuk memahami kemampuan finansial nasabah.
        - *jumlah transaksi*: untuk menilai seberapa sering nasabah menggunakan layanan bank.
        - *skor kredit*: untuk mengevaluasi risiko kredit nasabah.

      Contoh 2: Anda mengerjakan proyek untuk memprediksi churn pelanggan di perusahaan telekomunikasi. Anda memerlukan data seperti lama berlangganan, frekuensi panggilan ke pusat layanan pelanggan, paket data yang digunakan, dan riwayat panggilan dengan deskripsi sebagai berikut:

        - *lama berlangganan*: untuk menilai seberapa lama pelanggan telah menggunakan layanan.
        - *frekuensi panggilan ke pusat layanan pelanggan*: untuk melihat masalah atau ketidakpuasan pelanggan.
        - *paket data yang digunakan*: untuk memahami penggunaan layanan.
        - *riwayat tagihan*: untuk menganalisis pola pembayaran pelanggan.
    - Periksa ketersediaan data dari sumber yang relevan, baik internal maupun eksternal.

Proyek: Analisis Faktor yang Mempengaruhi Biaya Hidup Mahasiswa
Fitur yang Dipilih dan Alasan Pemilihan

1. Usia – Mengetahui apakah ada pola gaya hidup yang berubah berdasarkan usia.
2. Jenis Kelamin – Menganalisis perbedaan gaya hidup antara laki-laki dan perempuan.
3. Alamat & Kota Tempat Tinggal – Melihat pengaruh lokasi terhadap preferensi gaya hidup.
4. Jenis Kendaraan – Mengetahui apakah kepemilikan kendaraan mempengaruhi kebiasaan mobilitas mahasiswa.
5. Pengeluaran BBM – Mengukur pola mobilitas dan pengaruhnya terhadap gaya hidup.
6. IPK – Meneliti hubungan antara akademik dan kebiasaan sosial mahasiswa.
7. Hobi – Mengetahui preferensi aktivitas mahasiswa dalam waktu luang.
8. Tinggi dan Berat Badan – Melihat keterkaitan antara gaya hidup dengan kebiasaan kesehatan dan pola makan.

Ketersediaan Data
Semua fitur tersedia dalam dataset DataSIB2E.csv, namun beberapa data mungkin perlu dibersihkan sebelum analisis lebih lanjut.

Tahap 2: Pengambilan Data dengan Tools yang Tepat

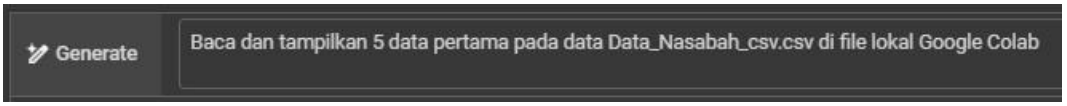
Langkah-langkah praktikum dalam pengambilan data dengan tools yang tepat

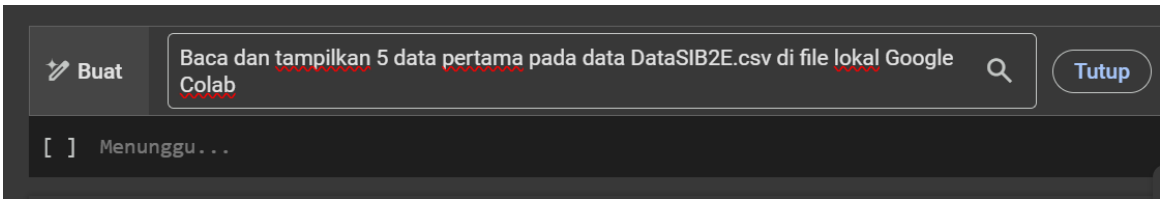
- Pilih tools dan metode pengambilan data yang sesuai. Gunakan tools seperti Python dengan pustaka BeautifulSoup untuk web scraping, atau API untuk mengambil data dari sumber yang tersedia secara online. Kaggle juga merupakan salah satu pilihan untuk pengambilan data.

  Berikut beberapa contoh cara mengerjakan sesuai dengan tools yang dipilih:

  - **Pengambilan data menggunakan *pandas* untuk file CSV**

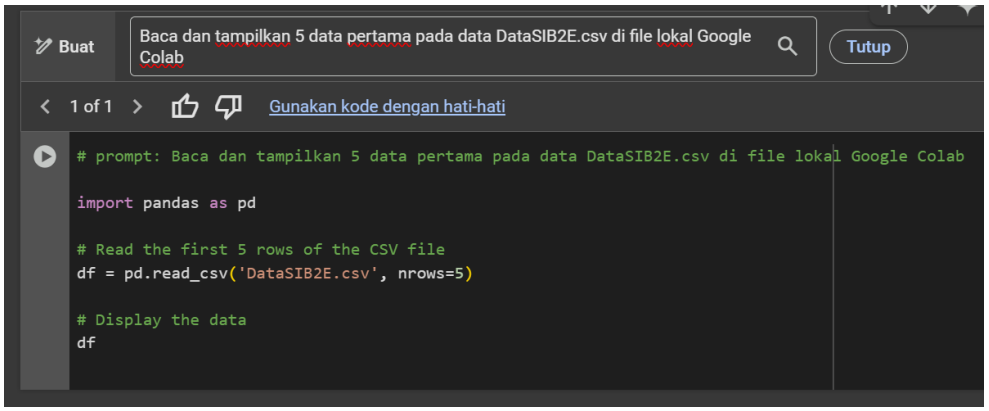
    Ketikan prompt pada *generate with AI* di Google Colab:





    Prompt tersebut akan menghasilkan kode seperti berikut:

```
import pandas as pd

# Baca file CSV lokal

data = pd.read_csv('/content/Data_Nasabah_csv.csv', delimiter=';')

# Tampilkan 5 data pertama

print(data.head(5))
```



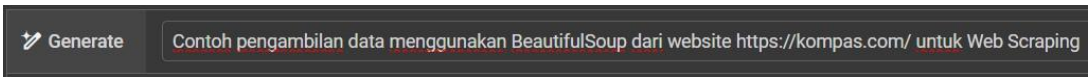
    Output dari kode yang dihasilkan akan tampak seperti di bawah ini:

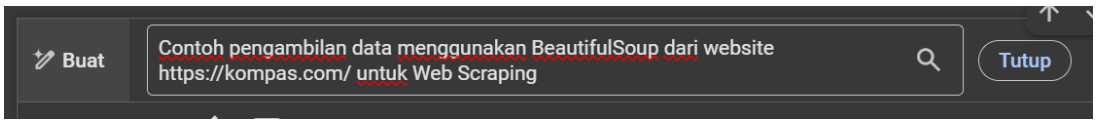
```
  nasabah_id  umur jenis_kelamin  pendapatan  saldo_rata_rata  \
0       N001    22     Perempuan     5800000          1508000
1       N002    64     Perempuan     5700000          1254000
2       N003    27     Perempuan     2950000           590000
3       N004    34     Perempuan     3100000           186000
4       N005    45     Laki-Laki     6700000          1474000

   jumlah_transaksi  jenis_produk  frekuensi_kunjungi_cabang  \
0                19      tabungan                          1
1                 9  kartu_kredit                          2
2                12      tabungan                          1
3                16      deposito                          5
4                15  kartu_kredit                          3

  pengguna_mobile_banking  skor_kredit
0                   TIDAK          900
1                   TIDAK          900
2                      YA          500
3                   TIDAK          700
4                      YA          800
```

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12
0	Absensi	Nama Lengkap	Tempat Tanggal Lahir	Usia	Jenis Kelamin	Alamat	Kota Tempat Tinggal	Jenis Kendaraan	Pengeluaran BBM	IPK	Hobi	Tinggi dan Berat Badan	Data Lain
1	1	Abhinaya Nuzuluzzuhdi	Malang, 31 Oktober 2004	20	Laki-laki	Jl. Gajayana	Malang	Sepeda Motor	Rp. 90.000	NaN	Maen Game	173/50kg	NaN
2	2	Alvi Choirinnikmah	Blitar, 9 September 2004	20	Perempuan	Jln. Kembang Turi	Malang	NaN	NaN	NaN	Menonton film	NaN	NaN
3	3	Alya Ajeng Ayu	Malang, 18 November 2004	20	Perempuan	Jl. Parkit Selatan no. 2	Malang	Sepeda Motor	Rp. 25.000 - 30.000	3.74	Baking	155cm 49kg	NaN
4	4	Ardhelia Putri Maharani	Malang, 11 Oktober 2004	20	Perempuan	Serenia Garden Regency B9	Malang	Sepeda Motor	Rp. 30.000	NaN	Live Tiktok	164 - 46	NaN

- **Pengambilan data menggunakan *BeautifulSoup* untuk Web Scraping**

  Ketikan prompt pada *generate with AI* di Google Colab:





Prompt tersebut akan menghasilkan kode seperti berikut:

```
import pandas as pd

import requests

from bs4 import BeautifulSoup


# URL yang ingin di-scrape

url = 'https://www.kompas.com/'


# Mengirim permintaan GET ke URL

response = requests.get(url)


# Memeriksa apakah permintaan berhasil

if response.status_code == 200:

  # Membuat objek BeautifulSoup dari konten HTML

  soup = BeautifulSoup(response.content,
'html.parser')


  # Mencari semua berita utama (misalnya, dengan tag
<h3 class="most__title">)

  berita_utama = soup.find_all('h3',
class_='most_title')


  # Menyimpan judul berita ke dalam list

  judul_berita = []

  for berita in berita_utama:

    judul_berita.append(berita.text.strip())


  # Membuat DataFrame pandas dari list judul berita

  df_berita = pd.DataFrame({'Judul Berita':
judul_berita})
```

```
    # Menampilkan DataFrame
    print(df_berita)


else:
    print(f'Gagal mengakses URL. Kode status: {response.status_code}')
```

```python
import pandas as pd

import requests

from bs4 import BeautifulSoup

# URL yang ingin di-scrape
url = 'https://www.kompas.com/'

# Mengirim permintaan GET ke URL
response = requests.get(url)

# Memeriksa apakah permintaan berhasil
if response.status_code == 200:

    # Membuat objek BeautifulSoup dari konten HTML
    soup = BeautifulSoup(response.content, 'html.parser')

    # Mencari semua berita utama (misalnya, dengan tag <h3 class="most__title">)
    berita_utama = soup.find_all('h3', class_='most__title')

    # Menyimpan judul berita ke dalam list
    judul_berita = []

    for berita in berita_utama:
        judul_berita.append(berita.text.strip())

    # Membuat DataFrame pandas dari list judul berita
    df_berita = pd.DataFrame({'Judul Berita': judul_berita})

    # Menampilkan DataFrame
    print(df_berita)

else:
    print(f'Gagal mengakses URL. Kode status: {response.status_code}')
```

Berikut untuk output kode di atas:

```
Empty DataFrame
Columns: [Judul Berita]
Index: []
```


```
Empty DataFrame
Columns: [Judul Berita]
Index: []
```

Oleh karena itu, butuh penyesuaian dengan melakukan inspect pada halaman website yang dipilih (dalam contoh ini Kompas) sebagai berikut:

```
import requests

from bs4 import BeautifulSoup

from datetime import datetime

import pytz


# URL yang ingin di-scrape

url = 'https://www.kompas.com/food/read/2024/10/08/103100275/resep-kukis-kacang-rendah-gula-cocok-untuk-pengidap-diabetes'


# Mengirim permintaan GET ke URL

response = requests.get(url)


# Memeriksa apakah permintaan berhasil

if response.status_code == 200:

    # Membuat objek BeautifulSoup dari konten HTML

    soup = BeautifulSoup(response.content, 'html.parser')
```

URL lebih spesifik

```
    # Mengambil judul artikel

    content_title = soup.find('h1').text    # Menyimpan
judul

    # Mencoba mengambil tanggal publikasi

    content_published_date = soup.find('meta',
property='article:published_time')

    if content_published_date:

        # Mengambil konten dan mengonversi ke objek
datetime

        utc_time =
datetime.fromisoformat(content_published_date['content
'].replace('Z', '+00:00'))



        # Mengonversi ke zona waktu WIB

        wib_time =
utc_time.astimezone(pytz.timezone('Asia/Jakarta'))

        published_date = wib_time.strftime('%d/%m/%Y,
%H:%M WIB')  # Format yang diinginkan

    else:

        published_date = "Tanggal publikasi tidak
ditemukan"



    # Mengambil tag

    content_tags = soup.find('meta', {'name':
'keywords'})

    if content_tags:

        tags = content_tags['content']     # Mengambil
tag dari tag meta

    else:

        tags = "Tag tidak ditemukan"




    # Menampilkan hasil
```

menggunakan h1 untuk menyimpan judul

```
        print("Judul Artikel:", content_title)
        print("Tanggal Publikasi:", published_date)
        print("Tag:", tags)

else:
    print(f'Gagal mengakses URL. Kode status: {response.status_code}')
```

Data yang ditampilkan lebih spesifik

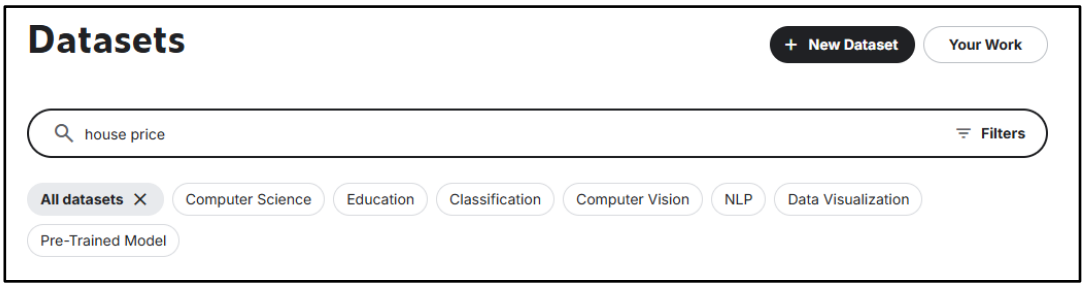
```python
import requests
from bs4 import BeautifulSoup
from datetime import datetime
import pytz

# URL yang ingin di-scrape
url = "https://www.kompas.com/food/read/2024/10/08/103100275/resep-kukis-kacang-rendah-gula-cocok-untuk-pengidap-diabetes"

# Menambahkan headers agar tidak diblokir
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36"
}

# Mengirim permintaan GET ke URL
response = requests.get(url, headers=headers)

# Memeriksa apakah permintaan berhasil
if response.status_code == 200:
    # Membuat objek BeautifulSoup dari konten HTML
    soup = BeautifulSoup(response.content, 'html.parser')

    # Mengambil judul artikel
    content_title = soup.find('h1').text.strip() if soup.find('h1') else "Judul tidak ditemukan"

    # Mencoba mengambil tanggal publikasi
    content_published_date = soup.find('meta', property='article:published_time')
    if content_published_date and 'content' in content_published_date.attrs:
        try:
            # Mengambil konten dan mengonversi ke objek datetime
            utc_time = datetime.fromisoformat(content_published_date['content'].replace('Z', '+00:00'))

            # Mengonversi ke zona waktu WIB
            wib_time = utc_time.astimezone(pytz.timezone('Asia/Jakarta'))
            published_date = wib_time.strftime('%d/%m/%Y, %H:%M WIB')  # Format yang diinginkan
        except ValueError:
            published_date = "Format tanggal tidak valid"
    else:
        published_date = "Tanggal publikasi tidak ditemukan"

    # Mengambil tag
    content_tags = soup.find('meta', {'name': 'keywords'})
    tags = content_tags['content'] if content_tags and 'content' in content_tags.attrs else "Tag tidak ditemukan"

    # Menampilkan hasil
    print("Judul Artikel:", content_title)
    print("Tanggal Publikasi:", published_date)
    print("Tag:", tags)

else:
    print(f'Gagal mengakses URL. Kode status: {response.status_code}')
```

Output yang akan diberikan sebagai berikut:

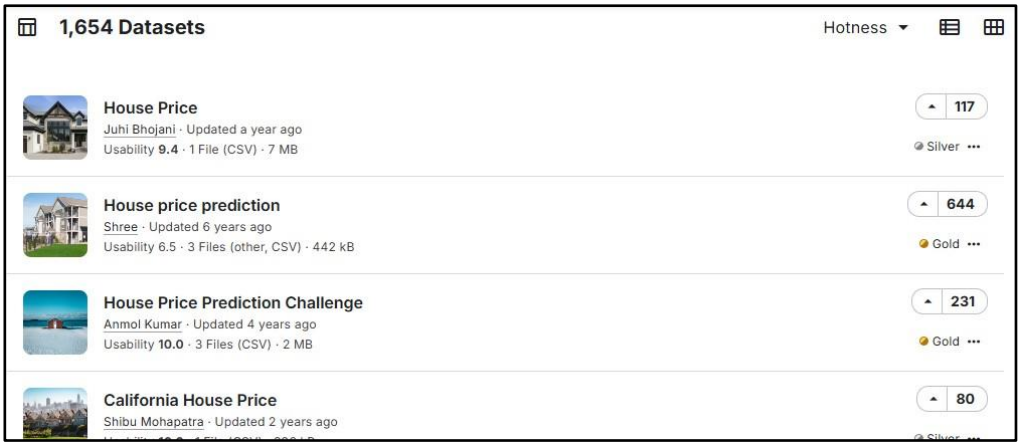
```
Judul Artikel: Resep Kukis Kacang Rendah Gula, Cocok untuk Pengidap Diabetes
Tanggal Publikasi: 08/10/2024, 10:31 WIB
Tag: Resep kukis kacang rendah gula, Resep kukis kacang rendah gula untuk pengidap diabetes, Resep makanan
```

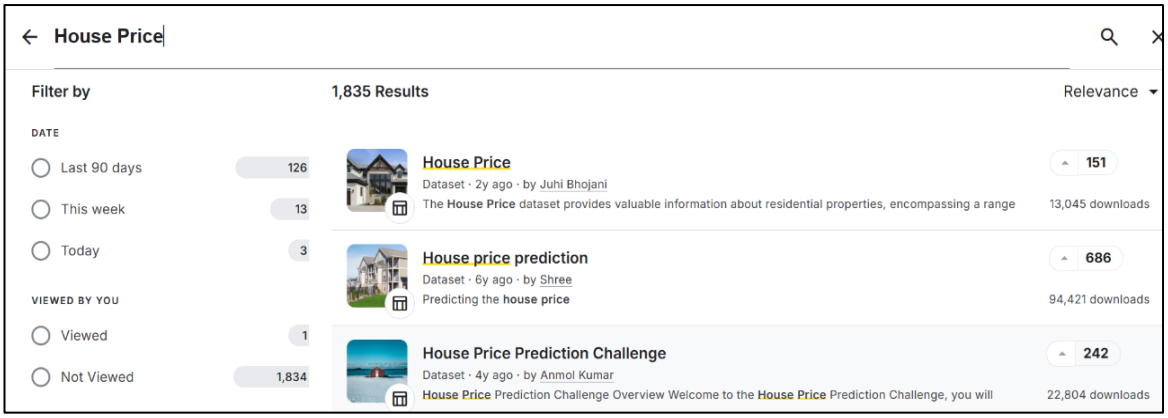
- **Pengambilan data melalui Kaggle**

Setelah login melalui website [Kaggle](), pilih bagian datasets dan ketik kata kunci data yang ingin dicari melalui search bar seperti pada gambar di bawah ini.

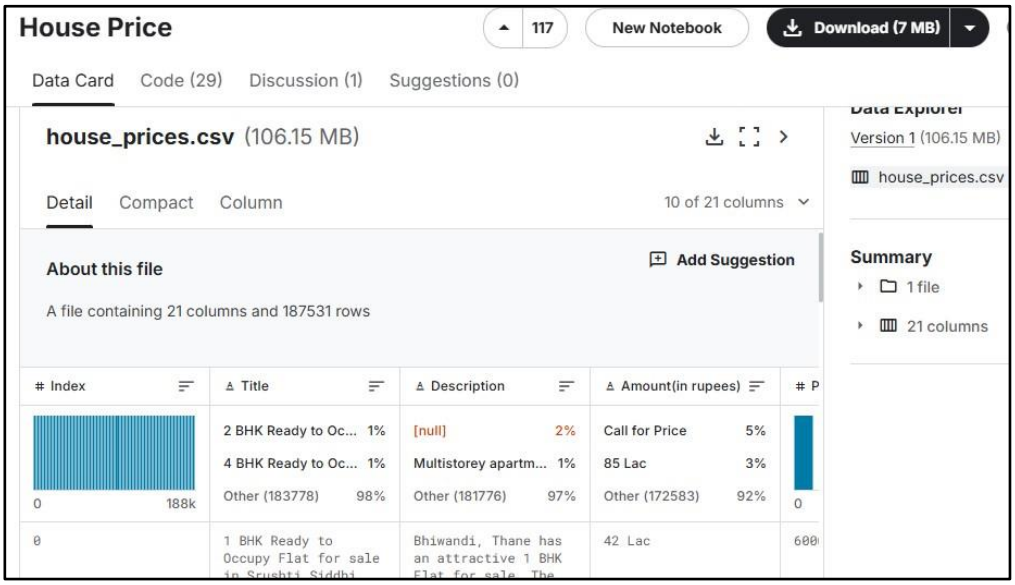


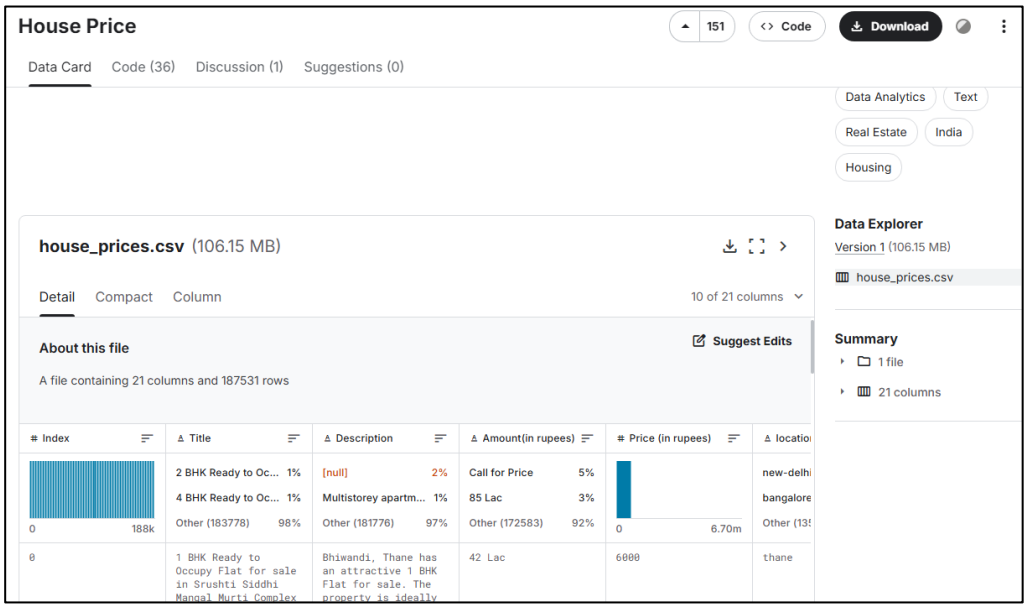
Kemudian pilih datasets yang relevan untuk proyek data science.



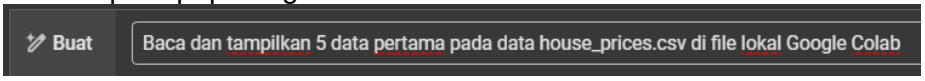


Lanjutkan dengan membaca deskripsi dan tekan tombol download untuk menyimpan dataset.





Ketikan prompt pada generate ai



Prompt akan menghasilkan kode

```
# prompt: Baca dan tampilkan 5 data pertama pada data house_prices.csv di file lokal Google Colab

import pandas as pd

# Assuming the file is named 'house_prices.csv' and is in the same directory as the notebook
try:
    df = pd.read_csv('house_prices.csv')
    print(df.head())
except FileNotFoundError:
    print("Error: 'house_prices.csv' not found. Please upload the file or provide the correct path.")
```

Unggah file dataset dan gunakan *pandas* untuk melihat isi dataset pada Google Colab seperti pada gambar di bawah ini.

Output yang di berikan

```
data = pd.read_csv('/content/house_prices.csv')

print(data.head())
```

```
   Index                                              Title  \
0      0  1 BHK Ready to Occupy Flat for sale in Srushti...
1      1  2 BHK Ready to Occupy Flat for sale in Dosti V...
2      2  2 BHK Ready to Occupy Flat for sale in Sunrise...
3      3        1 BHK Ready to Occupy Flat for sale Kasheli
4      4  2 BHK Ready to Occupy Flat for sale in TenX Ha...

                                         Description Amount(in rupees)  \
0  Bhiwandi, Thane has an attractive 1 BHK Flat f...           42 Lac
1  One can find this stunning 2 BHK flat for sale...           98 Lac
2  Up for immediate sale is a 2 BHK apartment in ...          1.40 Cr
3  This beautiful 1 BHK Flat is available for sal...           25 Lac
```

```
   Index                                              Title  \
0      0  1 BHK Ready to Occupy Flat for sale in Srushti...
1      1  2 BHK Ready to Occupy Flat for sale in Dosti V...
2      2  2 BHK Ready to Occupy Flat for sale in Sunrise...
3      3        1 BHK Ready to Occupy Flat for sale Kasheli
4      4  2 BHK Ready to Occupy Flat for sale in TenX Ha...

                                         Description Amount(in rupees)  \
0  Bhiwandi, Thane has an attractive 1 BHK Flat f...           42 Lac
1  One can find this stunning 2 BHK flat for sale...           98 Lac
2  Up for immediate sale is a 2 BHK apartment in ...          1.40 Cr
3  This beautiful 1 BHK Flat is available for sal...           25 Lac
4  This lovely 2 BHK Flat in Pokhran Road, Thane ...          1.60 Cr

   Price (in rupees) location Carpet Area         Status         Floor  \
0             6000.0    thane    500 sqft  Ready to Move  10 out of 11
1            13799.0    thane    473 sqft  Ready to Move   3 out of 22
2            17500.0    thane    779 sqft  Ready to Move  10 out of 29
3                NaN    thane    530 sqft  Ready to Move    1 out of 3
4            18824.0    thane    635 sqft  Ready to Move  20 out of 42

  Transaction  ... facing            overlooking  \
0      Resale  ...    NaN                    NaN
1      Resale  ...   East            Garden/Park
2      Resale  ...   East            Garden/Park
3      Resale  ...    NaN                    NaN
4      Resale  ...   West  Garden/Park, Main Road

                            Society Bathroom Balcony Car Parking  \
0  Srushti Siddhi Mangal Murti Complex        1       2         NaN
1                        Dosti Vihar        2     NaN      1 Open
2                  Sunrise by Kalpataru        2     NaN   1 Covered
3                                 NaN        1       1         NaN
4           TenX Habitat Raymond Realty        2     NaN   1 Covered

              Ownership Super Area Dimensions  Plot Area
0                   NaN        NaN        NaN        NaN
1             Freehold        NaN        NaN        NaN
2             Freehold        NaN        NaN        NaN
3                   NaN        NaN        NaN        NaN
4  Co-operative Society        NaN        NaN        NaN

[5 rows x 21 columns]
```

Tahap 3: Integrasi Data ke dalam Sistem

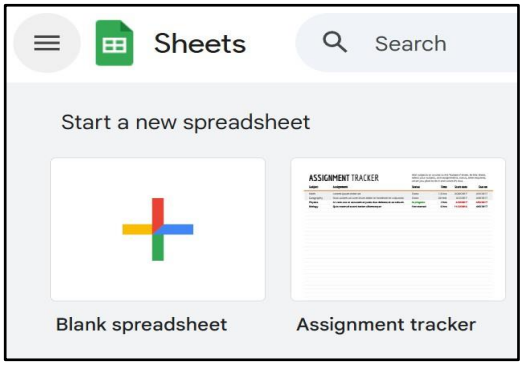
Langkah-langkah praktikum dalam integrasi data ke dalam sistem

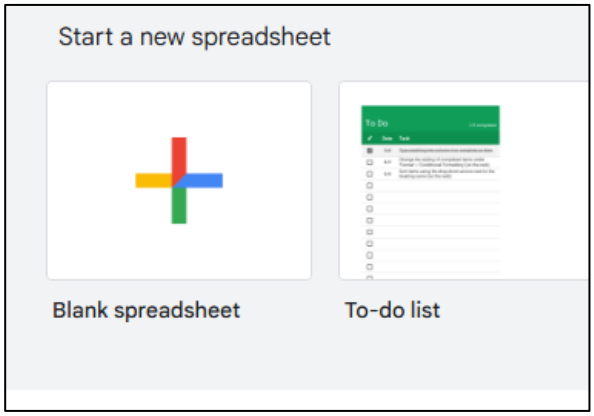
- Integrasikan data yang telah dibersihkan ke dalam database atau sistem penyimpanan data dengan memperhatikan beberapa hal berikut:
    - Gunakan tools basis data untuk menyimpan data, memastikan data diatur dengan baik dan dapat diakses untuk analisis di masa depan.
    - Verifikasi bahwa data terintegrasi dengan benar dan tidak ada kesalahan selama proses transfer.

  Berikut adalah contoh untuk integrasi data ke dalam sistem:

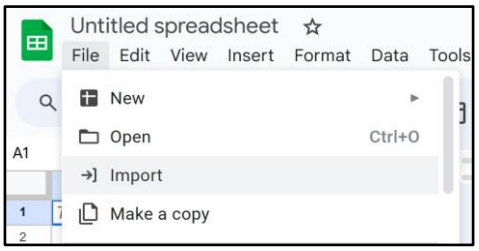
    - Menggunakan Google Spreadsheets

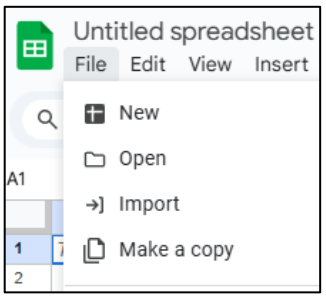
      Buka Google Spreadsheets dan pilih “Blank spreadsheet”



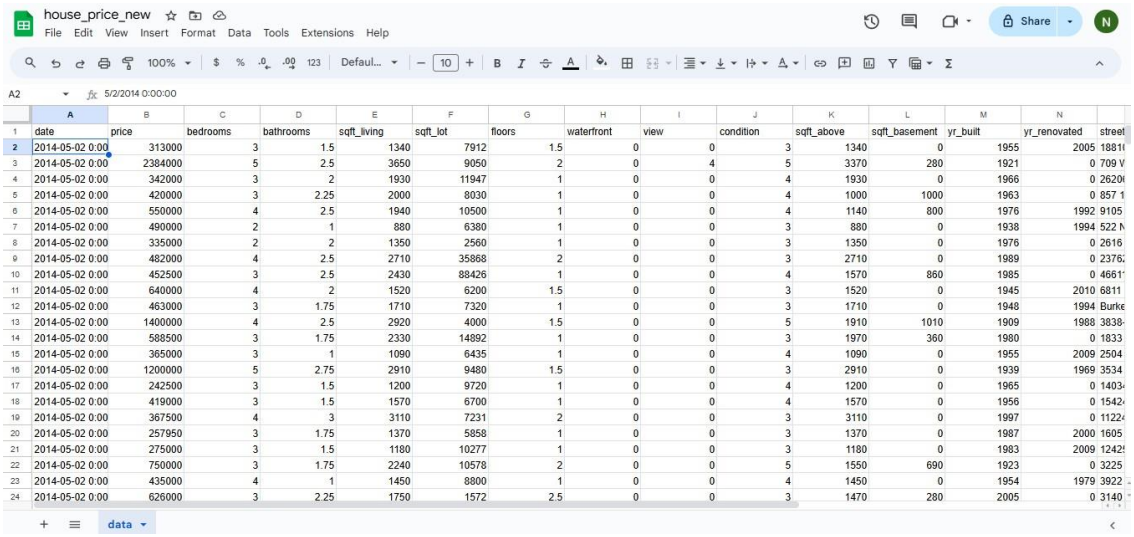


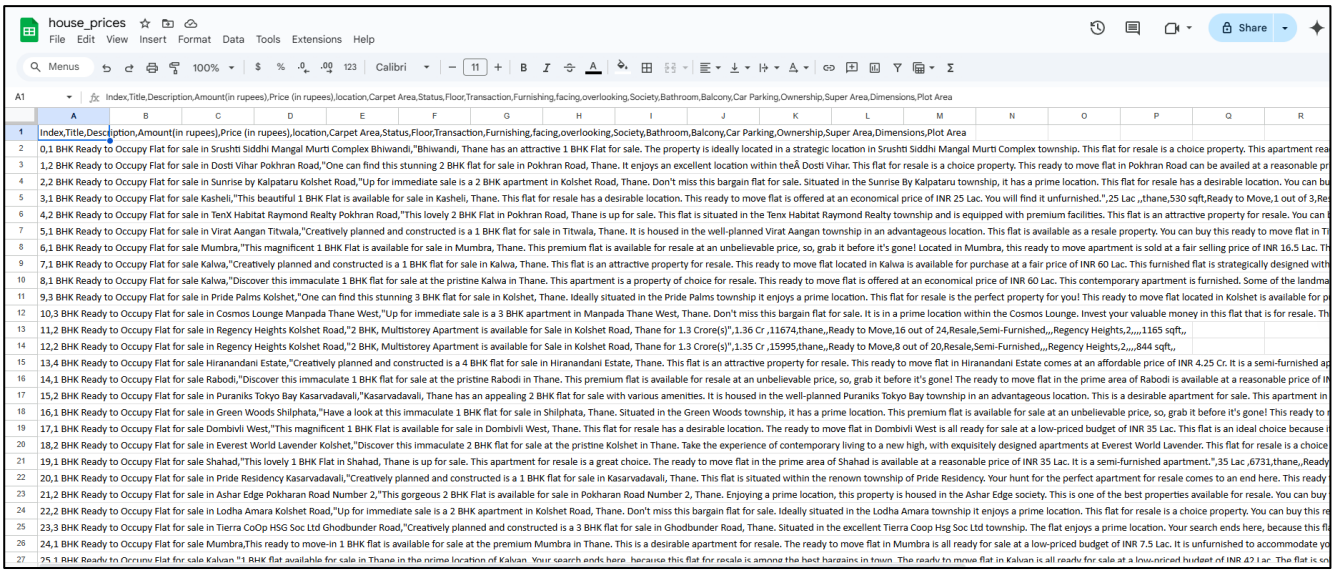
Pada menu File pilih “Import” dan pilih file dataset yang sudah di download.





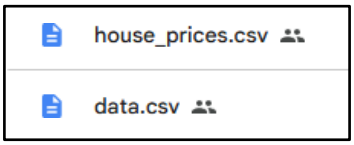
Beri nama file untuk dataset yang sudah diimport.

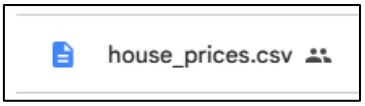




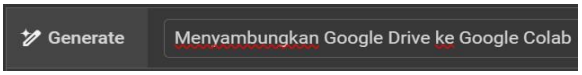
- Menggunakan Google Drive

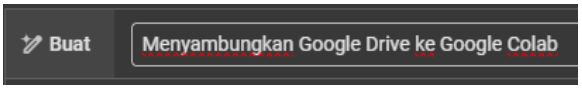
Pastikan file dataset sudah diupload di Google Drive.





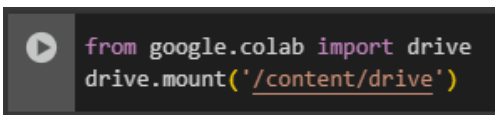
Buka Google Colab lalu ketikan prompt berikut:



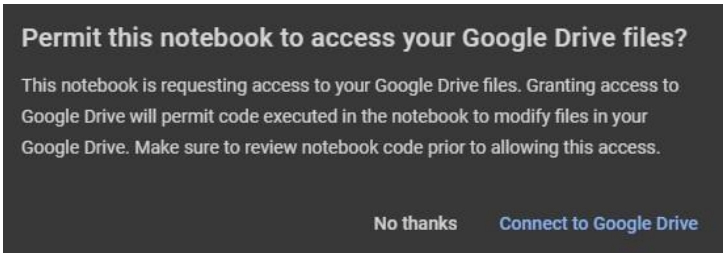


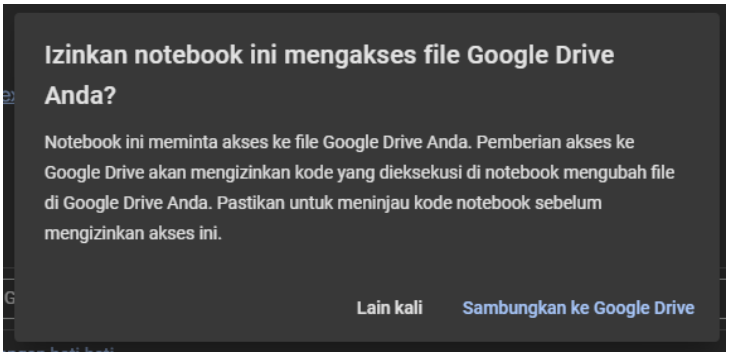
Lalu akan muncul kode berikut:

```
from google.colab import drive
drive.mount('/content/drive')
```

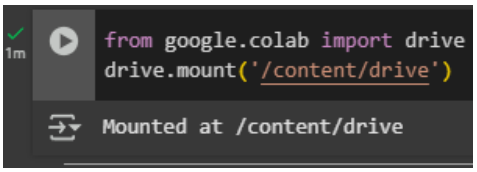


Setelah kode dijalankan akan muncul sebuah notifikasi, pilih “Connect to Google Drive”.

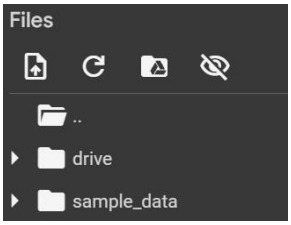


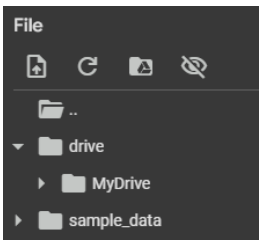


Berikut output jika sudah tersambung:

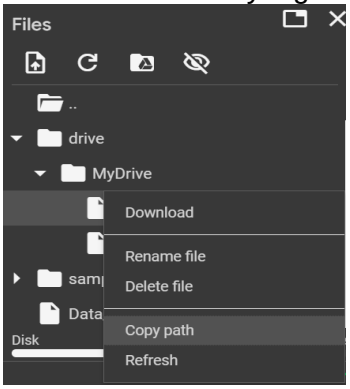


Pada menu Files akan muncul folder “drive”





Cari lokasi dataset yang telah diupload dan pilih “copy path”.



Baca data menggunakan *pandas*.

```
data = pd.read_csv('/content/drive/MyDrive/data.csv')

print(data.head())
```

```
                 date      price  bedrooms  bathrooms  sqft_living  sqft_lot
0  2014-05-02 00:00:00   313000.0       3.0       1.50         1340      7912
1  2014-05-02 00:00:00  2384000.0       5.0       2.50         3650      9050
2  2014-05-02 00:00:00   342000.0       3.0       2.00         1930     11947
3  2014-05-02 00:00:00   420000.0       3.0       2.25         2000      8030
4  2014-05-02 00:00:00   550000.0       4.0       2.50         1940     10500

   floors  waterfront  view  condition  sqft_above  sqft_basement  yr_built  \
0     1.5           0     0          3        1340              0      1955
1     2.0           0     4          5        3370            280      1921
2     1.0           0     0          4        1930              0      1966
3     1.0           0     0          4        1000           1000      1963
```

Buat

Baca dan tampilkan 5 data pertama pada data house_prices.csv di dalam Google Drive

Tutup

1 of 1

Gunakan kode dengan hati-hati

```
# prompt: Baca dan tampilkan 5 data pertama pada data house_prices.csv di dalam Google Drive

import pandas as pd

# Path to your CSV file in Google Drive
file_path = '/content/drive/MyDrive/house_prices.csv'  # Replace with your actual file path

try:
  # Read the CSV file into a pandas DataFrame
  df = pd.read_csv(file_path)

  # Display the first 5 rows of the DataFrame
  print(df.head())

except FileNotFoundError:
  print(f"Error: File not found at {file_path}")
except pd.errors.EmptyDataError:
  print(f"Error: The file at {file_path} is empty.")
except pd.errors.ParserError:
  print(f"Error: Unable to parse the file at {file_path}. Check the file format.")
except Exception as e:
  print(f"An unexpected error occurred: {e}")
```

```
   Index                                              Title  \
0      0  1 BHK Ready to Occupy Flat for sale in Srushti...
1      1  2 BHK Ready to Occupy Flat for sale in Dosti V...
2      2  2 BHK Ready to Occupy Flat for sale in Sunrise...
3      3        1 BHK Ready to Occupy Flat for sale Kasheli
4      4  2 BHK Ready to Occupy Flat for sale in TenX Ha...

                                         Description Amount(in rupees)  \
0  Bhiwandi, Thane has an attractive 1 BHK Flat f...              42 Lac
1  One can find this stunning 2 BHK flat for sale...              98 Lac
2  Up for immediate sale is a 2 BHK apartment in ...             1.40 Cr
3  This beautiful 1 BHK Flat is available for sal...              25 Lac
4  This lovely 2 BHK Flat in Pokhran Road, Thane ...             1.60 Cr

   Price (in rupees) location Carpet Area         Status         Floor  \
0             6000.0    thane    500 sqft  Ready to Move  10 out of 11
1            13799.0    thane    473 sqft  Ready to Move   3 out of 22
2            17500.0    thane    779 sqft  Ready to Move  10 out of 29
3                NaN    thane    530 sqft  Ready to Move    1 out of 3
4            18824.0    thane    635 sqft  Ready to Move  20 out of 42

  Transaction  ... facing            overlooking  \
0      Resale  ...    NaN                    NaN
1      Resale  ...   East         Garden/Park
2      Resale  ...   East         Garden/Park
3      Resale  ...    NaN                    NaN
4      Resale  ...   West  Garden/Park, Main Road

                                Society Bathroom Balcony Car Parking  \
0  Srushti Siddhi Mangal Murti Complex        1       2         NaN
1                          Dosti Vihar        2     NaN      1 Open
2                 Sunrise by Kalpataru        2     NaN  1 Covered
3                                  NaN        1       1         NaN
4           TenX Habitat Raymond Realty        2     NaN  1 Covered

              Ownership Super Area Dimensions  Plot Area
0                   NaN        NaN        NaN        NaN
1             Freehold         NaN        NaN        NaN
2             Freehold         NaN        NaN        NaN
3                   NaN        NaN        NaN        NaN
4  Co-operative Society        NaN        NaN        NaN

[5 rows x 21 columns]
```

Tugas Praktikum

1. **Identifikasi Kebutuhan Data**
   - Buat daftar fitur yang diperlukan untuk proyek data science yang kalian pilih, jelaskan alasan pemilihan setiap fitur, dan periksa ketersediaan data tersebut.
   - **Tugas**: Jelaskan secara singkat mengapa setiap fitur penting untuk tujuan proyek Anda.
2. **Pengambilan Data Menggunakan Kaggle dan Tools Lain**
   - Lakukan pengambilan data menggunakan tools yang telah dipelajari (misalnya, Python, API data, Kaggle).
   - **Tugas**: Serahkan skrip yang digunakan untuk mengambil data beserta hasilnya. Tulis deskripsi singkat tentang cara kerja skrip tersebut dan bagaimana data diambil.
3. **Integrasi Data ke dalam Database atau Repository**
   - Unggah data ke dalam database atau sistem penyimpanan data.
   - **Tugas**: Dokumentasikan proses integrasi data, termasuk tools yang digunakan dan verifikasi bahwa data terintegrasi dengan benar.

Jawaban

1. **Identifikasi Kebutuhan Data**

   Proyek: Analisis Penyebaran COVID-19 di Indonesia

   Fitur yang Dibutuhkan:

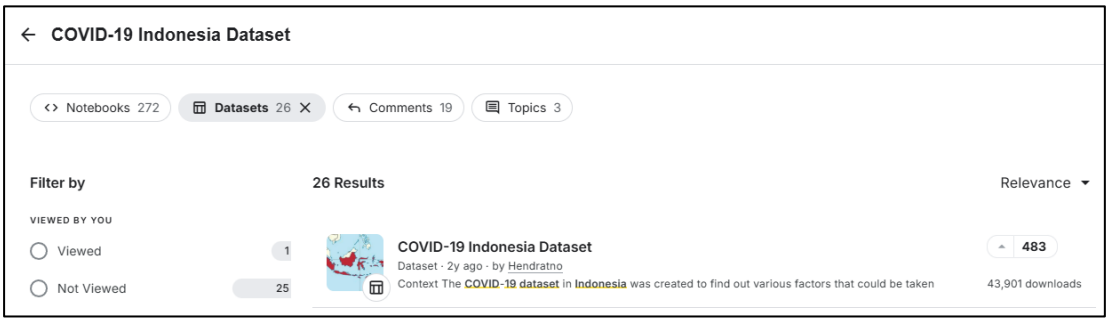
   1. Tanggal - Untuk melihat perkembangan kasus dari waktu ke waktu.
   2. Provinsi - Untuk menganalisis distribusi penyebaran COVID-19 di berbagai daerah.
   3. Jumlah Kasus Positif - Indikator utama untuk melihat tren penyebaran virus.
   4. Jumlah Kasus Sembuh - Mengetahui tingkat pemulihan pasien COVID-19.
   5. Jumlah Kasus Meninggal - Untuk menganalisis tingkat fatalitas akibat COVID-19.
   6. Jumlah Tes COVID-19 - Melihat efektivitas dan cakupan pengujian di Indonesia

   Ketersediaan Data:

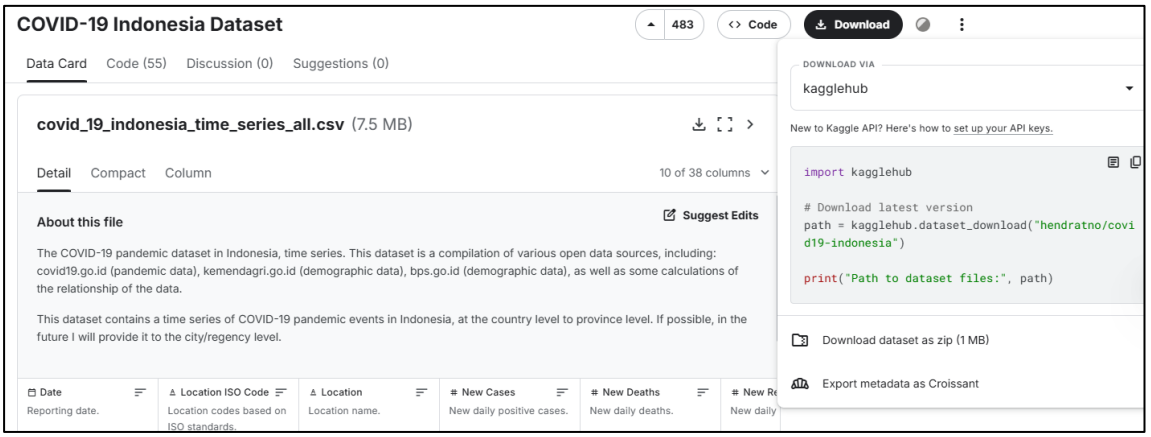
   Dataset *covid_19_indonesia.csv* yang telah diunggah berisi informasi yang mencakup fitur-fitur di atas.

2. **Pengambilan Data Menggunakan Kaggle**

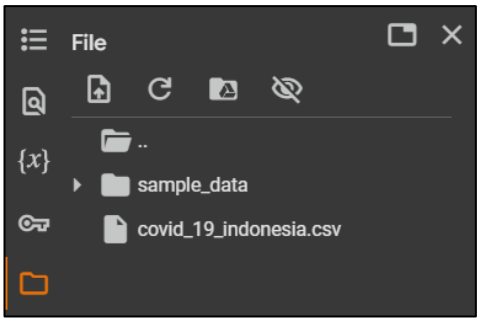
   1. Masuk ke Kaggle dan cari dataset terkait COVID-19 di Indonesia.



2. Unduh dataset dalam format CSV dari halaman dataset yang dipilih.

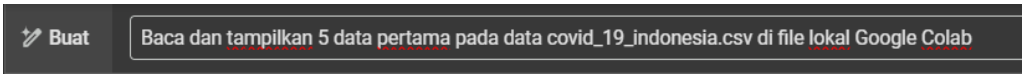


3. Unggah dataset ke Google Colab dan melakukan pengolahan data lainnya untuk diproses lebih lanjut.

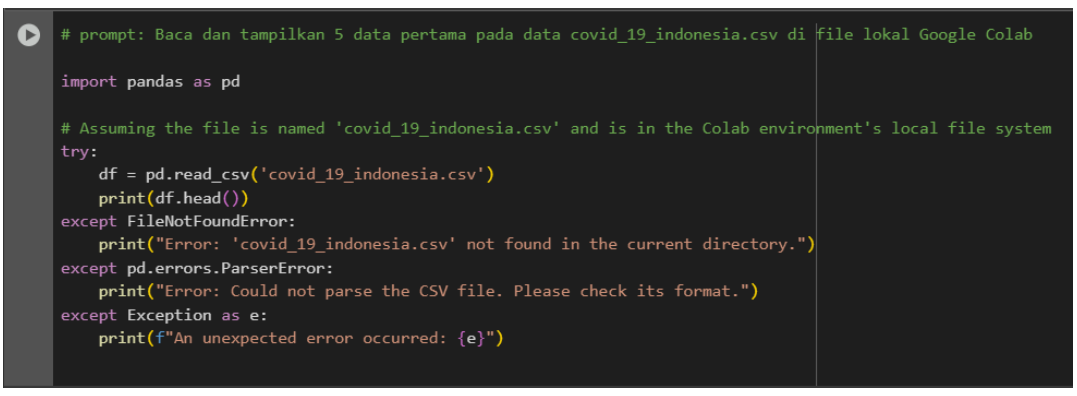


4. Gunakan Pandas untuk membaca dataset dalam format CSV dan melakukan eksplorasi awal data.

- Mengetikkan prompt pada generate with AI di Google Colab



- Menghasilkan kode berikut



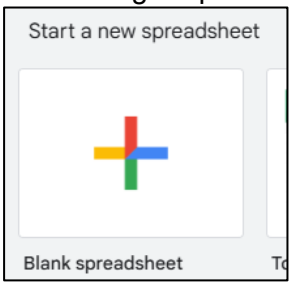
```
# prompt: Baca dan tampilkan 5 data pertama pada data covid_19_indonesia.csv di file lokal Google Colab

import pandas as pd

# Assuming the file is named 'covid_19_indonesia.csv' and is in the Colab environment's local file system
try:
    df = pd.read_csv('covid_19_indonesia.csv')
    print(df.head())
except FileNotFoundError:
    print("Error: 'covid_19_indonesia.csv' not found in the current directory.")
except pd.errors.ParserError:
    print("Error: Could not parse the CSV file. Please check its format.")
except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

- Menampilkan 5 data pertama menggunakan pandas untuk file CSV

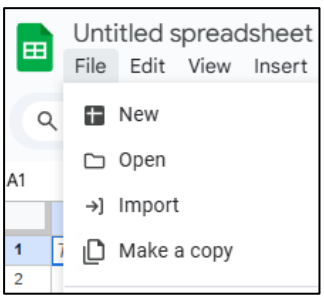
```
        Date Location ISO Code     Location  New Cases  New Deaths  \
0  3/1/2020            ID-JK  DKI Jakarta          2           0
1  3/2/2020            ID-JK  DKI Jakarta          2           0
2  3/2/2020              IDN    Indonesia          2           0
3  3/2/2020            ID-RI         Riau          1           0
4  3/3/2020            ID-JK  DKI Jakarta          2           0

   New Recovered  New Active Cases  Total Cases  Total Deaths  \
0              0                 2           39            20
1              0                 2           41            20
2              0                 2            2             0
3              0                 1            1             0
4              0                 2           43            20

   Total Recovered  ...  Latitude New Cases per Million  \
0               75  ... -6.204699                  0.18
1               75  ... -6.204699                  0.18
2                0  ... -0.789275                  0.01
3                1  ...  0.511648                  0.16
4               75  ... -6.204699                  0.18

   Total Cases per Million New Deaths per Million Total Deaths per Million  \
0                     3.60                    0.0                     1.84
1                     3.78                    0.0                     1.84
2                     0.01                    0.0                     0.00
3                     0.16                    0.0                     0.00
4                     3.96                    0.0                     1.84

  Total Deaths per 100rb Case Fatality Rate Case Recovered Rate  \
0                   0.18            51.28%             192.31%
1                   0.18            48.78%             182.93%
2                   0.00             0.00%               0.00%
3                   0.00             0.00%             100.00%
4                   0.18            46.51%             174.42%

  Growth Factor of New Cases  Growth Factor of New Deaths
0                        NaN                          NaN
1                        1.0                          1.0
2                        NaN                          NaN
3                        NaN
4                        1.0

[5 rows x 38 columns]
```

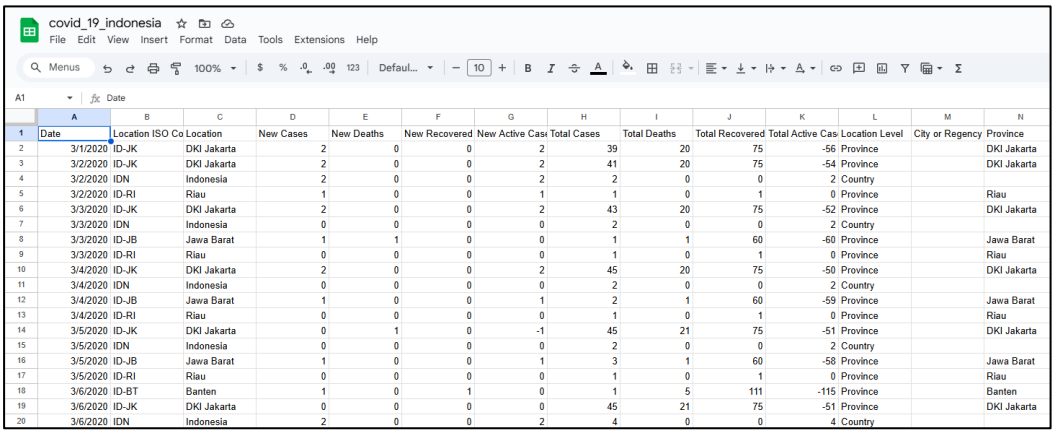
3. Integrasi Data ke dalam Sistem

1. Menggunakan Spreadsheets
   Buka Google Spreadsheets dan pilih “Blank spreadsheet”



2. Mengimport data set





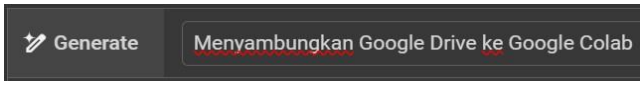
covid_19_indonesia

File Edit View Insert Format Data Tools Extensions Help

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Date	Location ISO Co	Location	New Cases	New Deaths	New Recovered	New Active Case	Total Cases	Total Deaths	Total Recovered	Total Active Cas	Location Level	City or Regency	Province
2	3/1/2020	ID-JK	DKI Jakarta	2	0	0	2	39	20	75	-56	Province		DKI Jakarta
3	3/2/2020	ID-JK	DKI Jakarta	2	0	0	2	41	20	75	-54	Province		DKI Jakarta
4	3/2/2020	IDN	Indonesia	2	0	0	2	2	0	0	2	Country		
5	3/2/2020	ID-RI	Riau	1	0	0	1	1	0	1	0	Province		Riau
6	3/3/2020	ID-JK	DKI Jakarta	2	0	0	2	43	20	75	-52	Province		DKI Jakarta
7	3/3/2020	IDN	Indonesia	0	0	0	0	2	0	0	2	Country		
8	3/3/2020	ID-JB	Jawa Barat	1	1	0	0	1	1	60	-60	Province		Jawa Barat
9	3/3/2020	ID-RI	Riau	0	0	0	0	1	0	1	0	Province		Riau
10	3/4/2020	ID-JK	DKI Jakarta	2	0	0	2	45	20	75	-50	Province		DKI Jakarta
11	3/4/2020	IDN	Indonesia	0	0	0	0	2	0	0	2	Country		
12	3/4/2020	ID-JB	Jawa Barat	1	0	0	1	2	1	60	-59	Province		Jawa Barat
13	3/4/2020	ID-RI	Riau	0	0	0	0	1	0	1	0	Province		Riau
14	3/5/2020	ID-JK	DKI Jakarta	0	1	0	-1	45	21	75	-51	Province		DKI Jakarta
15	3/5/2020	IDN	Indonesia	0	0	0	0	2	0	0	2	Country		
16	3/5/2020	ID-JB	Jawa Barat	1	0	0	1	3	1	60	-58	Province		Jawa Barat
17	3/5/2020	ID-RI	Riau	0	0	0	0	1	0	1	0	Province		Riau
18	3/6/2020	ID-BT	Banten	1	0	1	0	1	5	111	-115	Province		Banten
19	3/6/2020	ID-JK	DKI Jakarta	0	0	0	0	45	21	75	-51	Province		DKI Jakarta
20	3/6/2020	IDN	Indonesia	2	0	0	2	4	0	0	4	Country		

3. Menggunakan Google Drive
   Mengaploud data set ke Google Drive

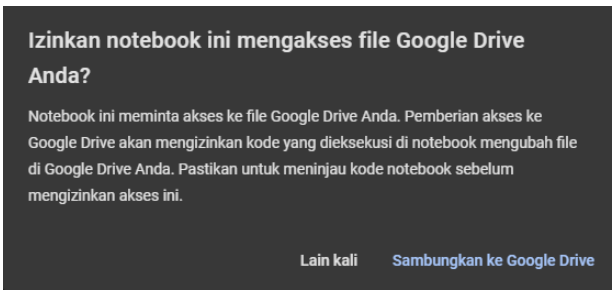
4. Buka Google Colab lalu ketikan prompt berikut



5. Lalu akan muncul kode berikut

```
from google.colab import drive
drive.mount('/content/drive')
```

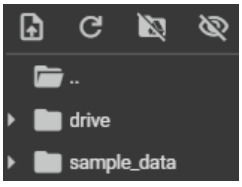
6. Setelah kode dijalankan akan muncul sebuah notifikasi, pilih “Connect to Google Drive”.



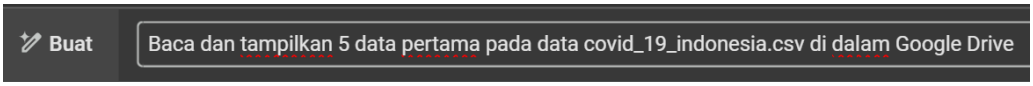
7. output jika sudah tersambung

```
Mounted at /content/drive
```

8. Pada menu Files akan muncul folder “drive”



9. Mengetikkan prompt pada generate with AI di Google Colab



10. Menghasilkan kode berikut

```
Kode yang dihasilkan mungkin tunduk pada lisensi |
# prompt: Baca dan tampilkan 5 data pertama pada data covid_19_indonesia.csv di dalam Google Drive

import pandas as pd

# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Read the CSV file from Google Drive
file_path = '/content/drive/My Drive/covid_19_indonesia.csv'  # Replace with the actual path
try:
    df = pd.read_csv(file_path)
    # Display the first 5 rows
    print(df.head())
except FileNotFoundError:
    print(f"Error: File not found at {file_path}")
except pd.errors.ParserError:
    print(f"Error: Unable to parse the CSV file at {file_path}. Please check the file format.")
except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

11. Menampilkan data awal

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
        Date Location ISO Code     Location  New Cases  New Deaths  \
0  3/1/2020            ID-JK  DKI Jakarta          2           0
1  3/2/2020            ID-JK  DKI Jakarta          2           0
2  3/2/2020              IDN    Indonesia          2           0
3  3/2/2020            ID-RI         Riau          1           0
4  3/3/2020            ID-JK  DKI Jakarta          2           0

   New Recovered  New Active Cases  Total Cases  Total Deaths  \
0              0                 2           39            20
1              0                 2           41            20
2              0                 2            2             0
3              0                 1            1             0
4              0                 2           43            20

   Total Recovered  ...  Latitude New Cases per Million  \
0               75  ... -6.204699                 0.18
1               75  ... -6.204699                 0.18
2                0  ... -0.789275                 0.01
3                1  ...  0.511648                 0.16
4               75  ... -6.204699                 0.18

   Total Cases per Million New Deaths per Million Total Deaths per Million  \
0                     3.60                    0.0                     1.84
1                     3.78                    0.0                     1.84
2                     0.01                    0.0                     0.00
3                     0.16                    0.0                     0.00
4                     3.96                    0.0                     1.84

  Total Deaths per 100rb Case Fatality Rate Case Recovered Rate  \
0                   0.18            51.28%             192.31%
1                   0.18            48.78%             182.93%
2                   0.00             0.00%               0.00%
3                   0.00             0.00%             100.00%
4                   0.18            46.51%             174.42%

  Growth Factor of New Cases  Growth Factor of New Deaths
0                        NaN                          NaN
1                        1.0                          1.0
2                        NaN                          NaN
3                        NaN                          NaN
4                        1.0                          1.0

[5 rows x 38 columns]
```