

Pedoman Menjalankan *Electronic Data Assistant (EDA)*

Versi

1. Node.js : Node 20.10.0
2. Python : Python 3.10.6
3. Flask : Flask 3.0.3
4. Streamlit : Streamlit 1.37.1
5. Langchain : Langchain 0.2.14

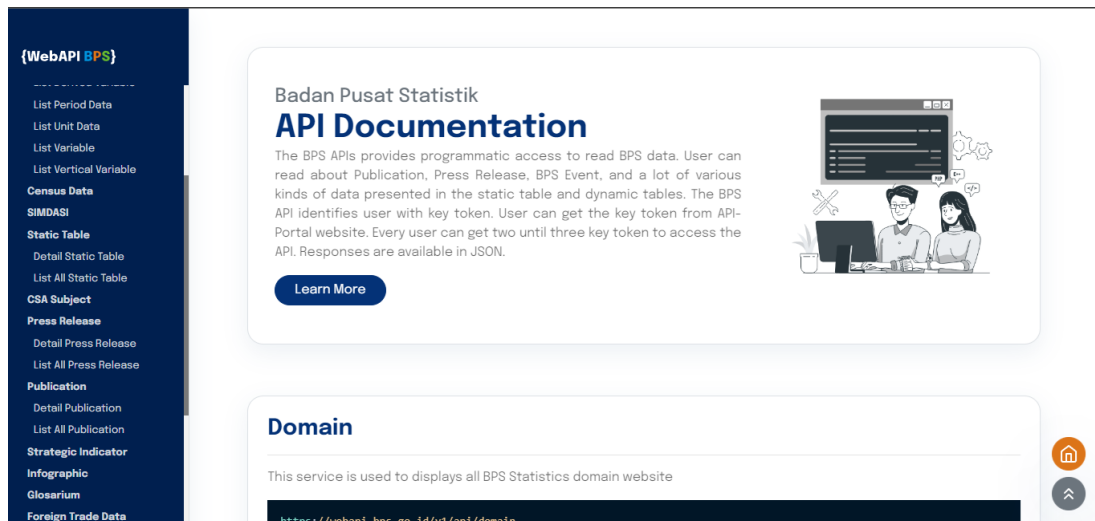
Persiapan

Sebelum menjalankan EDA, siapkan beberapa konfigurasi perangkat lunak dibawah ini sesuai dengan versi minimal yang diperlukan.

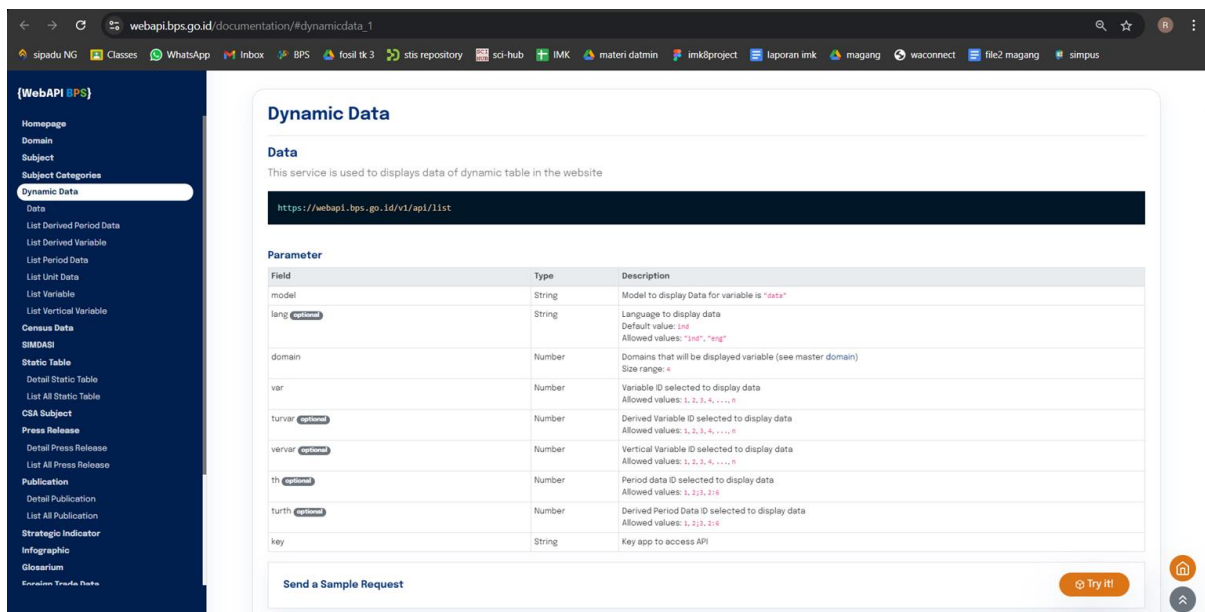
1. VSCode
Link install : <https://code.visualstudio.com/download>
Panduan : https://youtu.be/cu_ykIfBprI?si=hFRsIT5D9-_tv5It
2. Python
Link install : <https://www.python.org/downloads/>
Panduan : https://youtu.be/PfNFSwDianw?si=gc9PxZkS5Jpc_YQL
Lihat versi : `python --version`
3. Node.js
Link install : <https://nodejs.org/en>
Panduan : <https://youtu.be/06X51c6WHsQ?si=ai0K7NCf60jjo48a>
Install framework di terminal : `npm install express body-parser axios telegraf`
Lihat versi : `node -v`
4. Model
Install framework di terminal : `pip install flask langchain-community langchain-google-genai requests`
5. Streamlit
Install framework di terminal : `pip install pandas streamlit google-generativeai PyPDF2`
6. Postman
Link install : <https://www.postman.com/downloads/>
Panduan : https://youtu.be/5Fk6ef_mj9w?si=o0PiBfeyd8TKgAY-

A. Panduan untuk mendapatkan data dari Web API BPS

Data yang digunakan sebagai sumber informasi pada EDA diambil dari tabel Data Dinamis dari Web API BPS (<https://webapi.bps.go.id>). Anda harus memiliki akun PST terlebih dahulu untuk login ke dalam *website*. Masuk ke halaman dokumentasi dengan klik “Docs” di tab. Tampilan awal dokumentasi di Web API BPS sebagai berikut.



Anda dapat mencoba untuk mendapatkan berbagai data, seperti data dinamis, data sensus, SIMDASI, data publikasi, dan berbagai jenis data lainnya. Pada model EDA, Anda hanya membutuhkan data tabel dari data dinamis (*dynamic data*). Anda bisa mencoba di https://webapi.bps.go.id/documentation/#dynamicdata_1 atau *scroll* sampai *Dynamic Data*.



Lakukan percobaan pengambilan data dengan tombol *Try it* hingga muncul *pop up* berikut.


Data - This service is used to displays all variable on dynamic table in the website

<https://webapi.bps.go.id/v1/api/list/>

 Copy

model	<input type="text"/>
lang	<input type="text"/>
domain	<input type="text"/>
var	<input type="text"/>
turvar	<input type="text"/>
vervar	<input type="text"/>
th	<input type="text"/>
turth	<input type="text"/>

Masukkan **model** “data” dan **domain** sesuai dengan kode kantor BPS masing-masing, contohnya 1200 untuk BPS Provinsi Sumatera Utara. Isi **var** sesuai dengan ID data. Isi **key** yang merupakan **API Key** Anda. **API Key** Anda dapat diperoleh di tab “*Profile*” atau di [link https://webapi.bps.go.id/developer/user/profile](https://webapi.bps.go.id/developer/user/profile) sehingga akan muncul jendela berikut.


BADAN PUSAT STATISTIK


[Docs](#)
[Term of Use](#)
[Profile](#)
[Logout](#)

Accounts

[Profile](#)
[Applications](#)

Name	yo
Email	manuppak2002@gmail.com
Type Of Profession	Student
Name Of Company	STIS
Purpose Of Using Api	Others
Address Of Office	jakarta

Klik tab “*Applications*”, maka **API Key** Anda ada di kolom “*App ID*”.




BADAN PUSAT STATISTIK

[Docs](#)
[Term of Use](#)
[Profile](#)
[Logout](#)

Accounts

[Profile](#)
[Applications](#)

[Add Application](#)

No.	Title of Application	URL	Type of Application	Information	App ID	Status	
1	EDA	http://localhost/	Desktop Application	EDA	fc23698a6492a51c6833f4f39abe6f93	Aktif	 

<https://webapi.bps.go.id/developer/user/profile#application>

Silakan *copy* dan masukkan ke dalam *key*.

<https://webapi.bps.go.id/v1/api/115c7/model/data/domain/1200/var/50/key/1514sesuai1>

Copy

model

data

lang

domain

1200

var

50

turvar

vervar

th

turth

key

"isi sesuai key"

Send

Output:

Klik “*Send*” dan contoh hasilnya sebagai berikut.

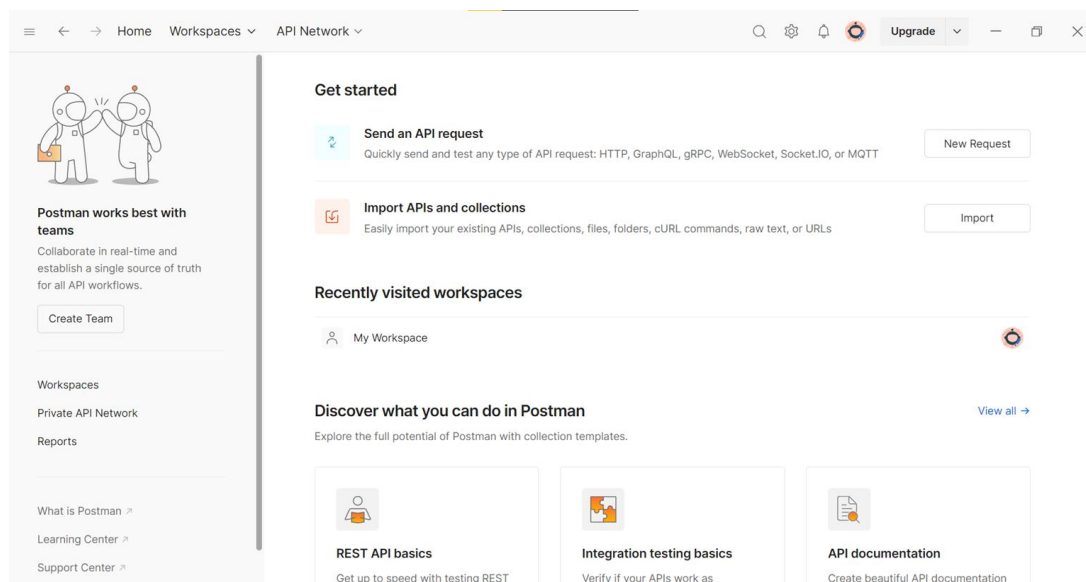
Output:

```
{
  "status": "OK",
  "data-availability": "available",
  "var": [
    {
      "val": 50,
      "label": "[Seri 2000]Produk Domestik Regional Bruto Atas Dasar Harga Konstan 2000 Menurut K",
      "unit": "Milyar Rupiah",
    }
  ]
}
```

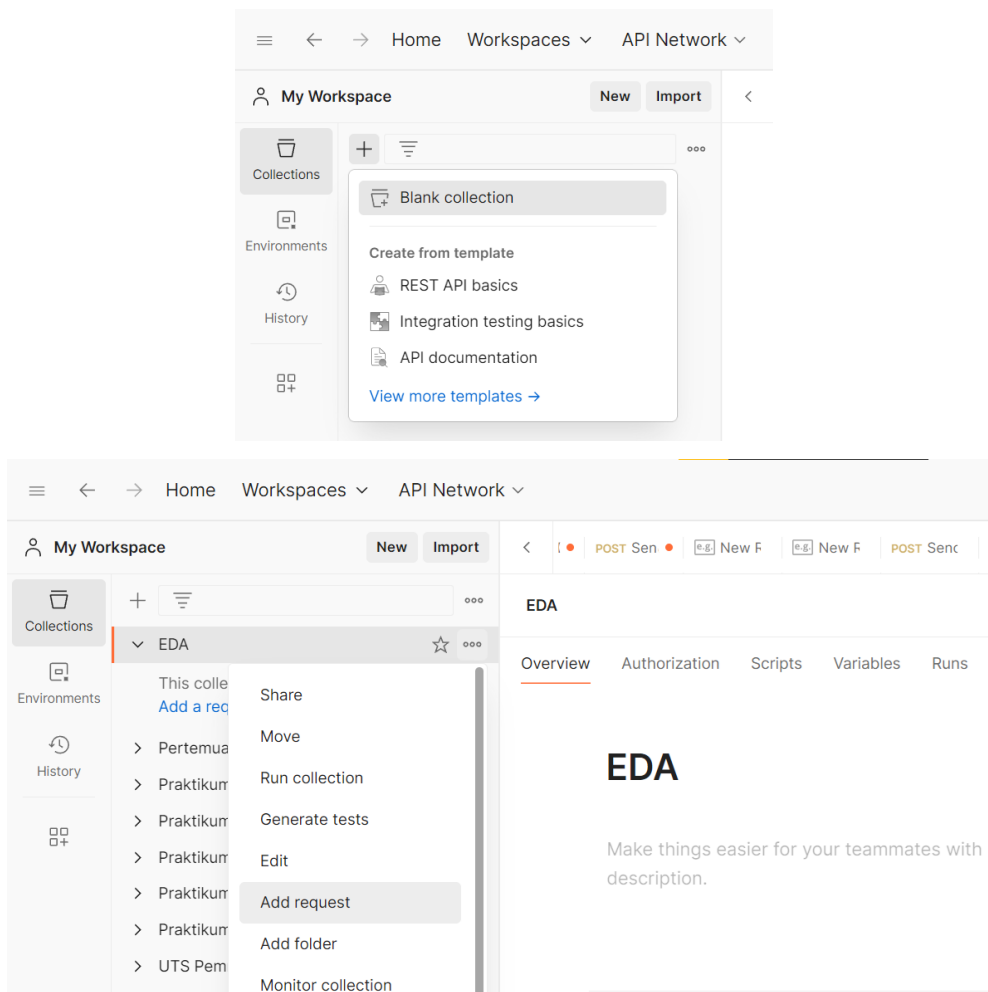
Lakukan hal yang sama untuk ID yang berbeda dan ambil data JSON sebanyak-banyaknya untuk dijadikan *dataset*.

B. Panduan untuk konversi JSON ke CSV

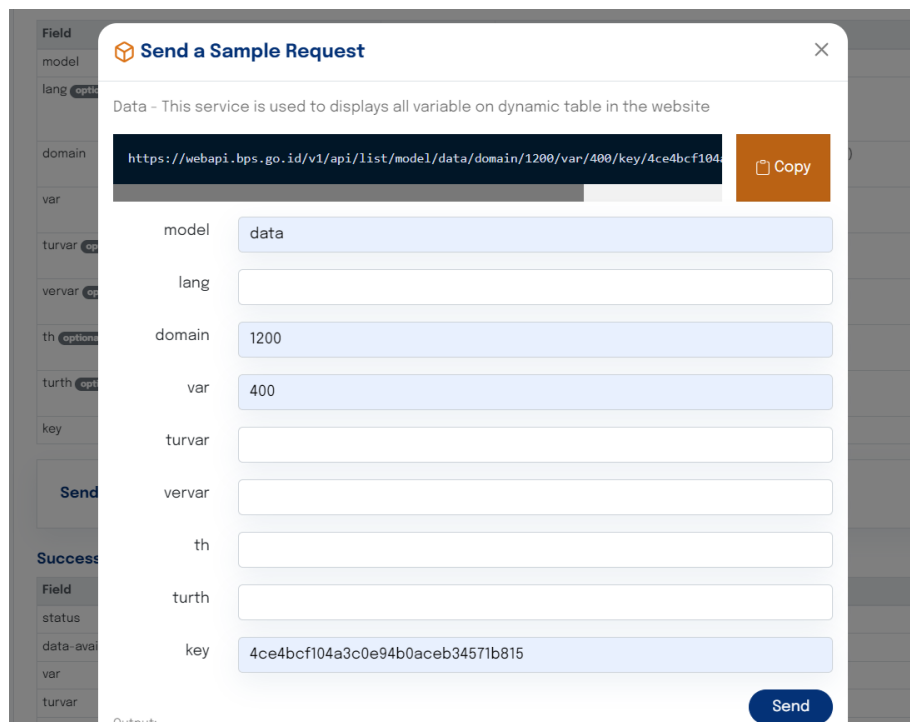
Agar kita dapat membaca data JSON, Anda perlu mengubah format data JSON menjadi berbentuk CSV. Buka aplikasi Postman dengan halaman Home sebagai berikut. Klik “My Workspace” untuk memulai *request*.



Klik *Create new collection* lalu *Blank collection*. Ubah nama *Collection* sesuai proyek, misalnya EDA. Pilih titik tiga di sebelah kanan *Collection* EDA, pilih *Add request*.



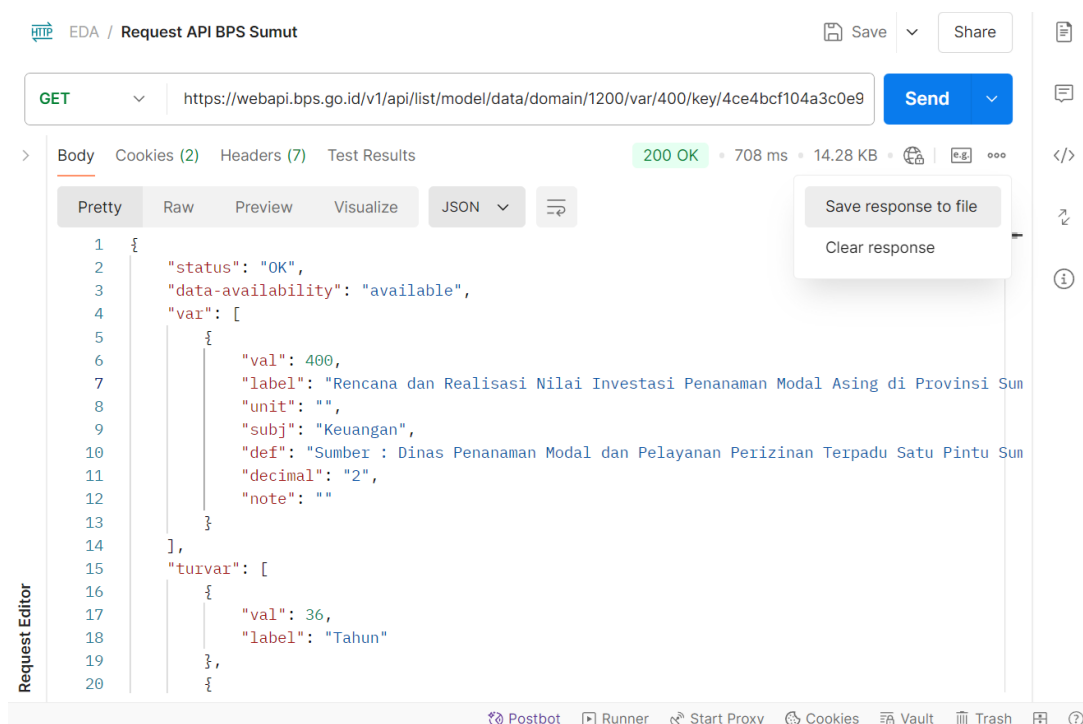
Copy url request API sesuai konfigurasi request pada website WebAPI BPS.



Adapun penjelasan dari tiap konfigurasi adalah sebagai berikut.

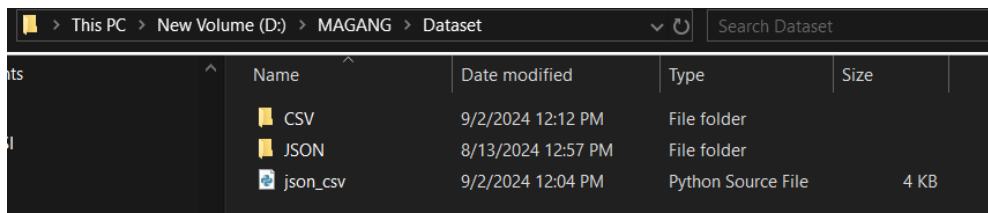
1. model : Untuk menampilkan data tabel, model harus diisi dengan **data**,
2. lang (opsional) : Pemilihan bahasa (secara default **ind** (bahasa Indonesia)), bisa diisi **ind** atau **eng** (bahasa Inggris)),
3. domain : asal daerah data yang akan diambil (diisi sesuai dengan nomor masing-masing BPS, misal 1200 untuk BPS Provinsi Sumatera Utara),
4. var (opsional) : ID dari data yang ingin diambil,
5. turvar (opsional) : variabel turunan dari data yang akan diambil,
6. vervar (opsional) : variabel kolom dari tabel data yang akan diambil,
7. th (opsional) : periode data yang akan diambil,
8. turth (opsional) : periode turunan data yang akan diambil
9. key : API Key Anda untuk mengambil data.

Paste url yang telah di-copy untuk melakukan *request* data JSON dengan metode *GET*. Klik *Send*, maka data JSON akan muncul. Terlihat pada BPS Provinsi Sumatera Utara dengan domain 1200 dan ID data ke-400 terdapat data Rencana dan Realisasi Nilai Investasi Penanaman Modal Asing di Provinsi Sumatera Utara menurut Sektor (ribu US\$). Klik *Save response to file* pada titik tiga untuk mengunduh data JSON ke *local computer* Anda. Simpan data JSON sesuai dengan direktori penyimpanan yang Anda inginkan.



Sebelumnya, pastikan Anda telah membuat folder Dataset/ yang berisi folder bernama

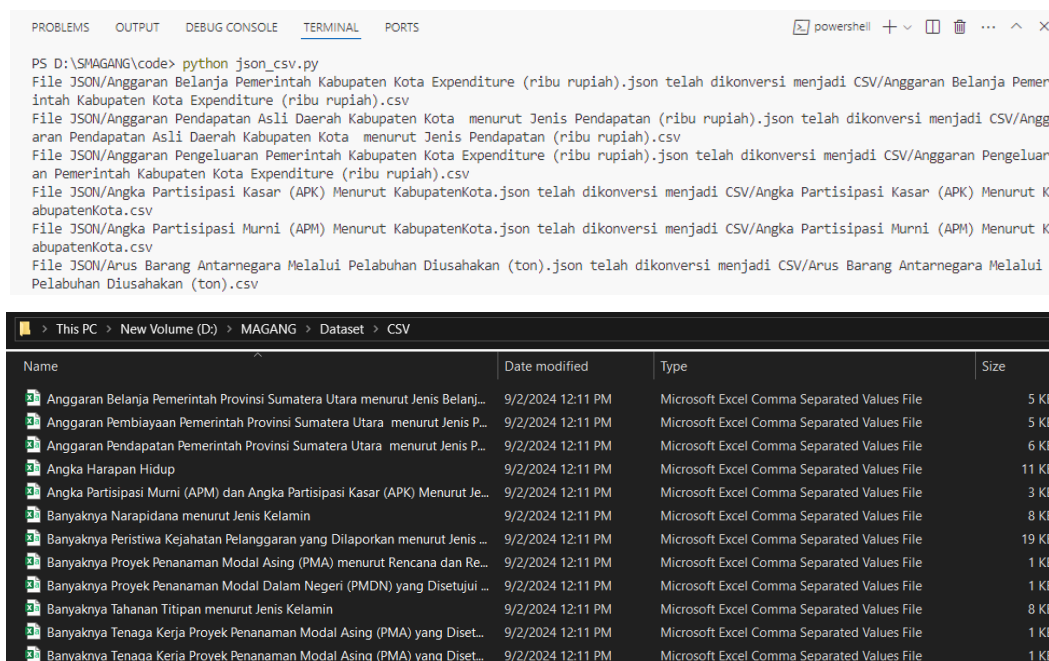
JSON/ dan CSV/. Anda menyimpan file JSON di folder JSON/ dan menyimpan file CSV hasil konversi dari file JSON di folder CSV/. Selain itu, simpan juga file **json_csv.py** di folder yang sama. Berikut contoh tampilan penyimpanannya.



Untuk mengubah format file JSON ke CSV, jalankan file **json_csv.py** di terminal VSCode atau cmd dengan direktori penyimpanan yang sama perintah `python json_csv.py`.



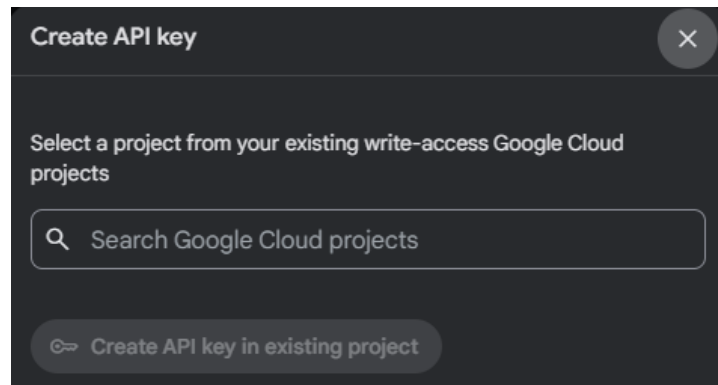
Setelah di-run, tampilan hasil konversi file CSV Anda sebagai berikut. Seluruh dataset JSON Anda akan otomatis berubah dan tersimpan ke folder CSV Anda.



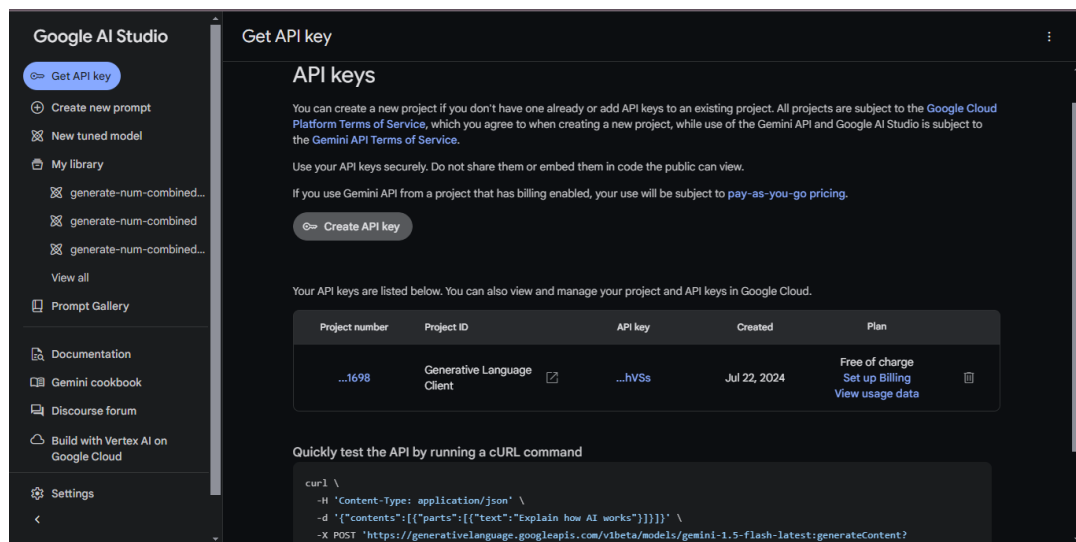
C. Panduan untuk mendapatkan API Google Key

Google API Key adalah token dari Google untuk bisa menggunakan layanan Google, salah satunya adalah Google Gemini AI.

1. Buka link <https://aistudio.google.com/app/apikey> yang harus sudah login.
2. Klik **Create API key** dan tampilan jendela baru akan muncul.



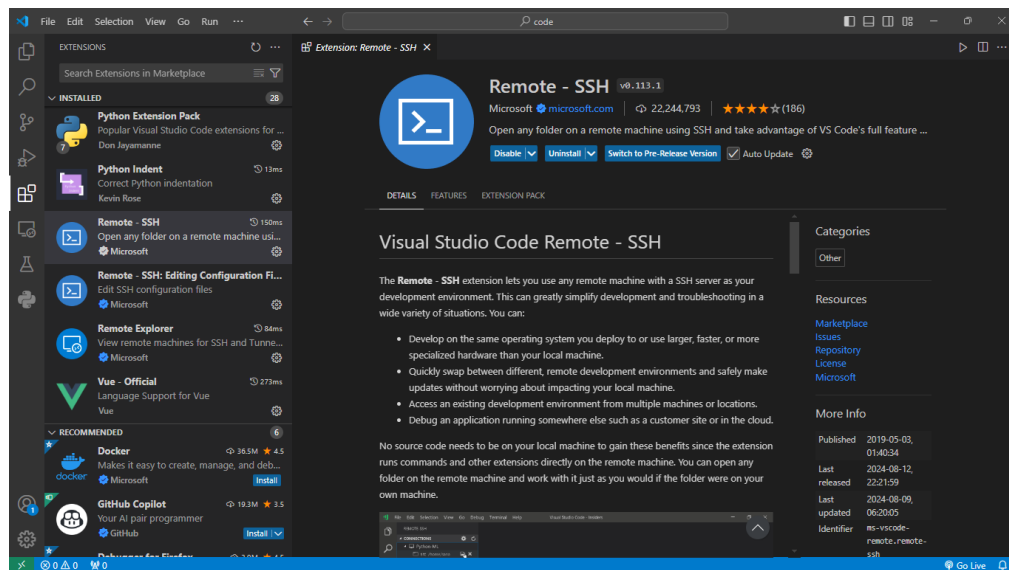
3. Jika pernah membuat proyek Google Cloud, Anda dapat memilih di pencarian dan Anda juga dapat membuat API key yang baru jika belum pernah membuat proyek Google Cloud sebelumnya.
4. Terdapat isian pada tabel untuk ID dari proyek dan API key.



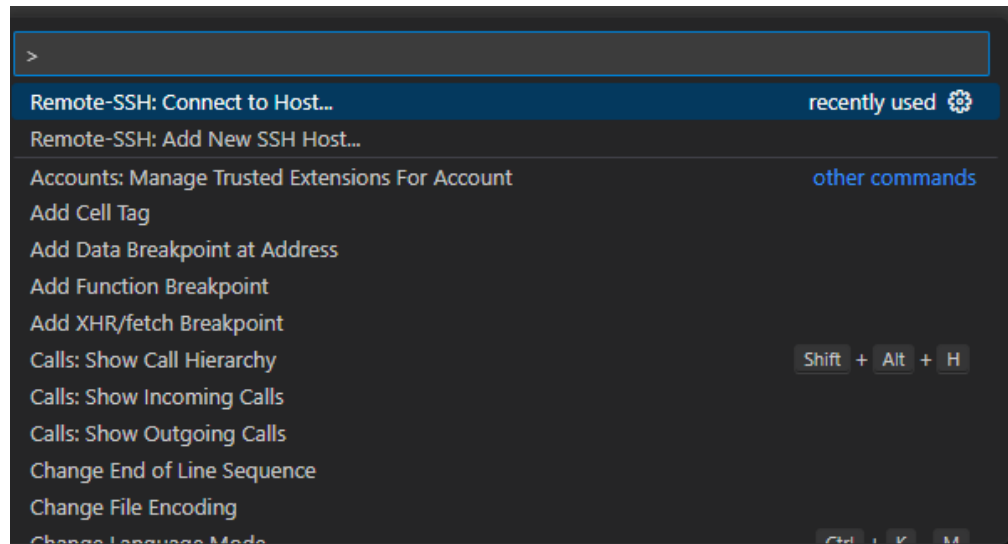
D. Panduan edit kodingan di server melalui VSCode :

1. Pastikan Anda telah meng-install VSCode.

2. Install ekstensi Remote – SSH.

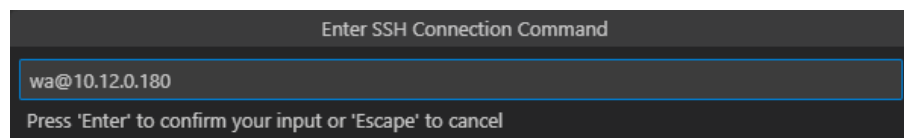


3. Buka **Command Palette** pada tab View atau dengan Ctrl + Shift + P.



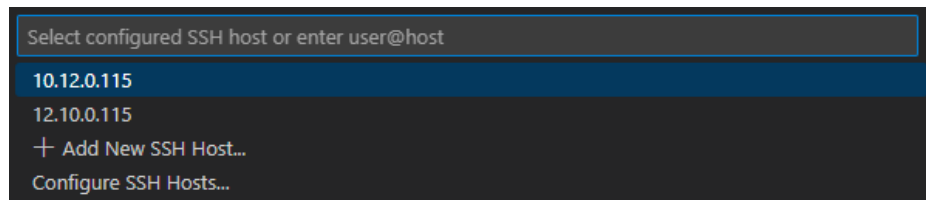
4. Pilih Host SSH yang baru (**Add New SSH Host**) dan masukkan nama_server@ip_address.

Contoh

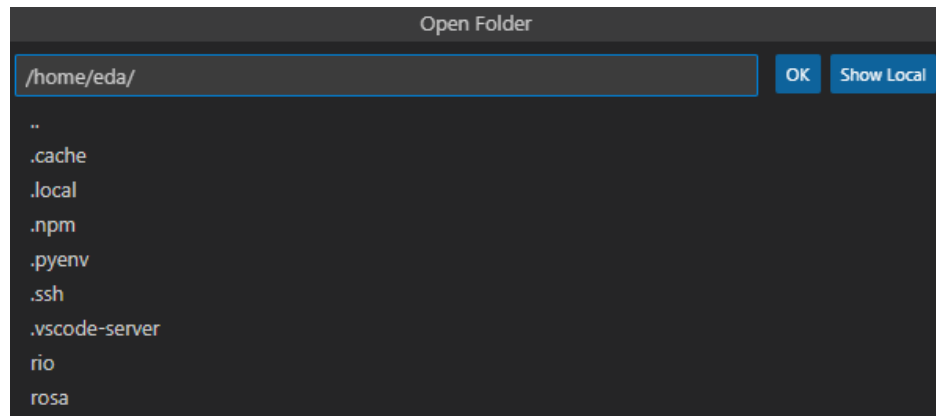


5. Masukkan password dan lanjutkan proses sampai selesai.

6. Pilih **Connect to Host** dan pilih ip address server.



7. Masukkan password untuk masuk.
8. Pilih folder yang dibuka.

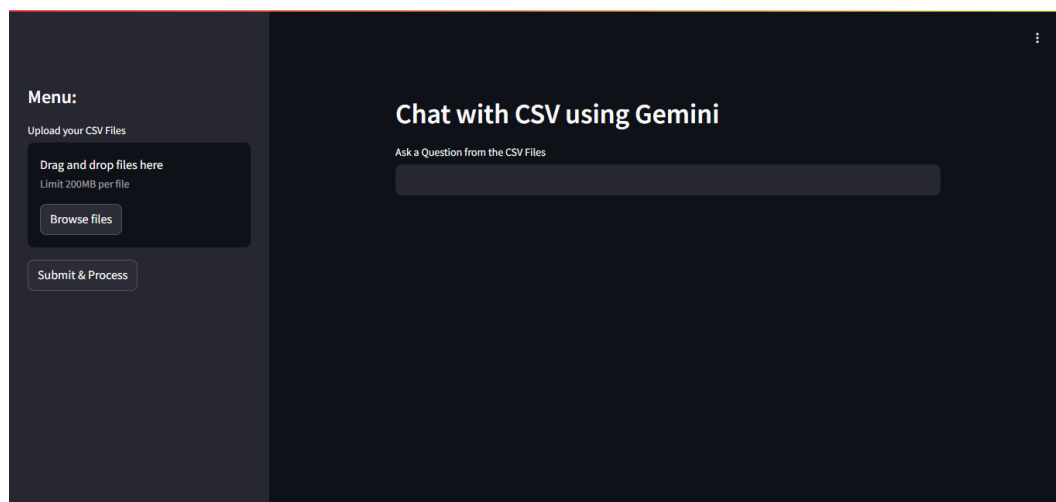


9. Masukkan password jika diminta.
10. Fitur edit langsung di server dapat dilakukan.

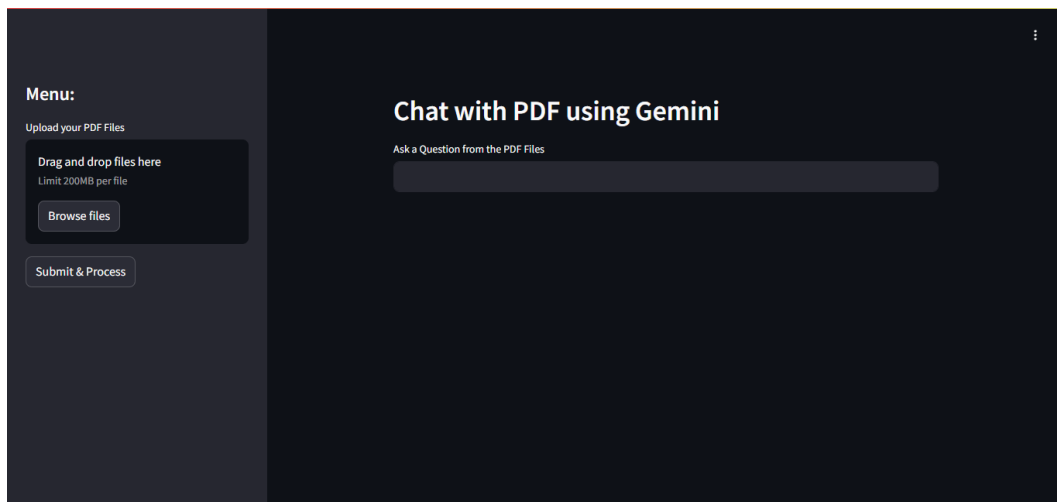
E. Entry data

Untuk memasukkan data yang akan dipelajari oleh model, Anda harus memasukkan data ke Streamlit baik Streamlit CSV atau Streamlit PDF. Berikut tampilan awal Streamlit.

Streamlit CSV



Streamlit PDF



1. Anda memasukkan file melalui Input Browse Files dan klik Submit & Process. Pastikan untuk memasukkan dengan selang 100 file tiap Submit & Process.
2. Tunggu sampai selesai.
3. Anda bisa menguji model apakah belajar atau tidak dari data yang Anda masukkan sebelumnya melalui input teks di Streamlit.

F. Panduan untuk menjalankan program :

1. Hosting seluruh kodingan **main_model.py**, **main_flask.py**, **main_wa.js**, **main_tele.js**.
2. Buat nomor untuk WA yang akan digunakan untuk EDA dan bot Telegram untuk EDA.
3. Untuk mendapatkan pesan dan mengirim pesan ke WA bisa mengirim alamat IP server dan nomor WA Anda untuk didaftarkan ke WAMConnect BPS Provinsi Sumatera Utara. Untuk pendaftaran dapat menghubungi *contact person* berikut :
Mas Didik : 081319303814
Rosalia : 081260551434
Rio : 082123042087
4. Masukkan API key nomor WA yang terdapat di WAMConnect, API key bot Telegram, dan Google API key yang digunakan untuk EDA satker Anda.
5. Untuk menjalankan python:
 - a. **python main_model.py** (di terminal untuk menjalankan model gemini),
 - b. **python main_flask.py** (di terminal untuk menjalankan flask).
6. Untuk menjalankan node js:
 - a. **node main_wa.js** (di terminal untuk menjalankan Node JS untuk WA),
 - b. **node main_tele.js** (di terminal untuk menjalankan Node JS untuk Telegram).

7. Untuk menjalankan Streamlit

- a. **streamlit run streamlit_read_csv.py** (di terminal untuk menjalankan Streamlit untuk input data csv),
- b. **streamlit run streamlit_read_pdf.py** (di terminal untuk menjalankan Streamlit untuk input data pdf).