

MANUAL BOOK

Klasifikasi Kualitas Udara Berdasarkan Data Sentinel-5P

Jak Air

Oleh:

Tanjaya Jason Winata
535220041

Pembimbing:

Prof. Dr. Ir. Dyah Erny Herwindiati, M.Si.
Janson Hendryli, S.Kom., M.kom.



DAFTAR ISI

DAFTAR ISI.....	2
PENDAHULUAN	3
CARA INSTALASI DAN SETUP APLIKASI <i>JAK AIR</i>	5
CARA PENGAMBILAN DATA SENTINEL-5P.....	8
CARA PENGGUNAAN APLIKASI <i>JAK AIR</i>	16

PENDAHULUAN

Aplikasi Jak Air merupakan hasil perancangan sistem klasifikasi kualitas udara spasial-temporal di wilayah DKI Jakarta dengan memanfaatkan data satelit Sentinel-5P dan algoritma *Support Vector Machine* (SVM). Aplikasi ini dikembangkan menggunakan framework Streamlit, sehingga dapat dijalankan secara interaktif melalui browser tanpa memerlukan instalasi tambahan selain Python dan pustaka pendukungnya.

Jak Air dirancang untuk membantu pengguna mendeteksi, mengklasifikasikan, dan memahami kondisi kualitas udara berdasarkan empat jenis polutan utama, yaitu karbon monoksida (CO), nitrogen dioksida (NO₂), sulfur dioksida (SO₂), dan ozon (O₃). Melalui antarmuka yang sederhana dan intuitif, aplikasi ini memungkinkan pengguna untuk melatih model SVM, melakukan pengujian data, serta menganalisis hasil klasifikasi secara visual menggunakan peta spasial dan grafik batang.

Selain berfungsi sebagai implementasi hasil perancangan, Jak Air juga dapat digunakan sebagai media pembelajaran dan eksplorasi interaktif, yang mempermudah pengguna memahami hubungan antara data satelit dan kondisi udara aktual di wilayah perkotaan. Aplikasi Jak Air terdiri dari enam modul utama yang diakses melalui menu navigasi pada sidebar. Setiap modul memiliki fungsi yang saling berkaitan dalam alur kerja sistem. Berikut penjelasan dari masing-masing modul:

1. Halaman Utama
Halaman pertama yang muncul saat aplikasi dijalankan. Berisi judul dan deskripsi singkat mengenai tujuan aplikasi, yaitu klasifikasi kualitas udara berbasis data satelit Sentinel-5P dengan algoritma SVM. Halaman ini berfungsi sebagai pengantar sebelum pengguna masuk ke fitur utama.
2. Halaman Pelatihan dan Pengujian
Modul ini merupakan inti dari aplikasi. Pengguna dapat mengunggah dataset Sentinel-5P, mengatur parameter SVM (seperti kernel, C, dan gamma), serta melatih model secara mandiri. Setelah proses pelatihan selesai, pengguna dapat melakukan pengujian menggunakan data baru untuk memperoleh hasil klasifikasi. Selain itu, halaman ini juga menampilkan perbandingan performa antar kernel (Linear dan RBF) sehingga pengguna dapat memilih model dengan hasil terbaik.
3. Halaman Hasil
Menampilkan hasil klasifikasi model yang telah dilatih dan diuji. Data hasil disajikan dalam bentuk peta spasial dan grafik batang agar pengguna dapat memahami distribusi serta kategori kualitas udara secara visual. Halaman ini membantu pengguna menganalisis kondisi udara secara keseluruhan dengan tampilan yang interaktif dan informatif.
4. Halaman Informasi
Berisi penjelasan mengenai empat polutan utama, mencakup sumber, dampak, serta karakteristiknya. Selain itu, halaman ini juga menjelaskan metode klasifikasi SVM yang digunakan, termasuk kernel Linear dan RBF. Pengguna dapat berpindah antar topik polutan maupun metode dengan tombol interaktif.

5. Halaman Panduan

Menyediakan petunjuk penggunaan aplikasi secara lengkap, mulai dari fungsi setiap modul, langkah pelatihan dan pengujian model, hingga interpretasi hasil visualisasi. Panduan ini disusun agar pengguna baru dapat memahami alur penggunaan aplikasi dengan mudah dan menjalankan setiap fitur tanpa kesulitan.

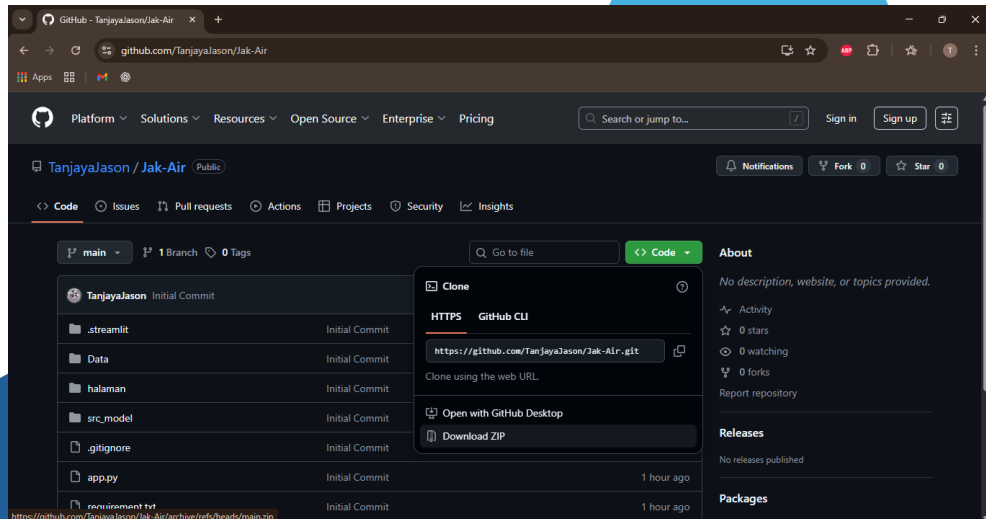
6. Halaman Tentang Pengembang

Memuat profil pengembang aplikasi, termasuk nama, identitas akademik, serta dosen pembimbing. Bagian ini juga menjelaskan latar belakang perancangan aplikasi, yang berangkat dari meningkatnya isu polusi udara di DKI Jakarta dan kebutuhan akan sistem pemantauan berbasis data satelit yang lebih efisien dan terbuka.

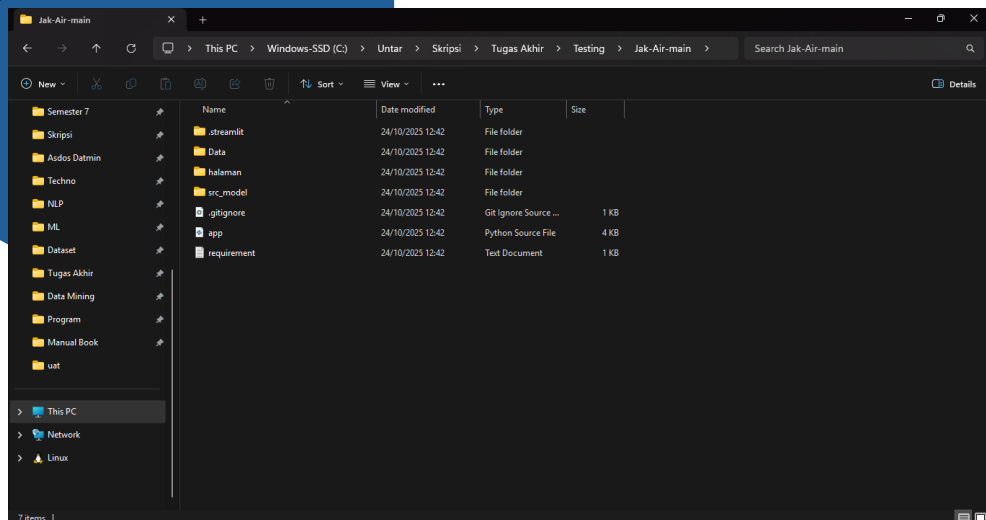
CARA INSTALASI DAN SETUP APLIKASI JAK AIR

Berikut adalah langkah-langkah untuk melakukan instalasi dan setup aplikasi Sistem Klasifikasi Kualitas Udara Jak Air pada komputer secara lokal:

1. Buka repositori Jak Air melalui link <https://github.com/TanjayaJason/Jak-Air.git> pada aplikasi web browser.
2. Klik tombol 'Code', kemudian pilih 'Download ZIP' untuk mengunduh seluruh file proyek.



3. Setelah proses unduhan selesai, ekstrak file tersebut ke dalam folder yang diinginkan.



4. Pastikan komputer sudah terpasang Python versi 3.9 atau lebih baru. Jika belum memiliki Python, unduh dan instal dari <https://www.python.org/downloads/>. Saat proses instalasi, centang opsi "Add Python to PATH" agar dapat digunakan langsung dari terminal.
5. Jalankan Visual Studio Code atau Command Prompt (CMD).
6. Arahkan ke folder proyek hasil ekstrak tadi dengan perintah `cd path\ke\folder_proyek`

```
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>cd C:\Untar\Skripsi\Tugas Akhir\Testing\Jak-Air-main

C:\Untar\Skripsi\Tugas Akhir\Testing\Jak-Air-main>
```

7. Buat Virtual Environment dan aktifkan environment tersebut. Pastikan muncul tulisan (venv) di awal baris terminal.

```
C:\Untar\Skripsi\Tugas Akhir\Testing\Jak-Air-main>python -m venv venv

C:\Untar\Skripsi\Tugas Akhir\Testing\Jak-Air-main>venv\Scripts\activate

(venv) C:\Untar\Skripsi\Tugas Akhir\Testing\Jak-Air-main>
```

8. Menginstall library yang dibutuhkan melalui file requirement.txt. Tunggu hingga semua dependensi selesai diinstall.

```
(venv) C:\Untar\Skripsi\Tugas Akhir\Testing\Jak-Air-main>pip install -r requirement.txt
Collecting streamlit
  Using cached streamlit-1.50.0-py3-none-any.whl (10.1 MB)
Collecting pandas
  Using cached pandas-2.3.3-cp311-cp311-win_amd64.whl (11.3 MB)
Collecting numpy
  Using cached numpy-2.3.4-cp311-cp311-win_amd64.whl (13.1 MB)
Collecting scipy
  Using cached scipy-1.16.2-cp311-cp311-win_amd64.whl (38.7 MB)
Collecting scikit-learn
  Using cached scikit_learn-1.7.2-cp311-cp311-win_amd64.whl (8.9 MB)
Collecting joblib
  Using cached joblib-1.5.2-py3-none-any.whl (308 kB)
Collecting matplotlib
  Using cached matplotlib-3.10.7-cp311-cp311-win_amd64.whl (8.1 MB)
Collecting seaborn
  Using cached seaborn-0.13.2-py3-none-any.whl (294 kB)
Collecting openpyxl
  Using cached openpyxl-3.1.5-py2.py3-none-any.whl (250 kB)
Collecting xlswriter
  Using cached xlswriter-3.2.9-py3-none-any.whl (175 kB)
Collecting geopandas
  Using cached geopandas-1.1.1-py3-none-any.whl (338 kB)
Collecting altair!=5.4.0,!=5.4.1,<6,>=4.0
  Using cached altair-5.5.0-py3-none-any.whl (731 kB)
Collecting blinker<2,>=1.5.0
  Using cached blinker-1.9.0-py3-none-any.whl (8.5 kB)
Collecting cachetools<7,>=4.0
  Using cached cachetools-6.2.1-py3-none-any.whl (10.1 kB)
Installing collected packages: pytz, xlswriter, watchdog, urllib3, tzdata, typing-extensions, tornado, toml, threadpoolctl, tenacity, smmap, six, r
pds-py, pyparsing, pyarrow, protobuf, pillow, packaging, numpy, narwhals, MarkupSafe, kiwisolver, joblib, idna, fonttools, et-xmlfile, cycler, color
ama, charset-normalizer, certifi, cachetools, blinker, attrs, shapely, scipy, requests, referencing, python-dateutil, pyproj, pyogrio, openpyxl, jin
ja2, gitdb, contourpy, click, scikit-learn, pydeck, pandas, matplotlib, jsonschema-specifications, gitpython, seaborn, jsonschema, geopandas, altair
, streamlit
Successfully installed MarkupSafe-3.0.3 altair-5.5.0 attrs-25.4.0 blinker-1.9.0 cachetools-6.2.1 certifi-2025.10.5 charset-normalizer-3.4.4 click-8.
3.0 colorama-0.4.6 contourpy-1.3.3 cycler-0.12.1 et-xmlfile-2.0.0 fonttools-4.60.1 geopandas-1.1.1 gitdb-4.0.12 gitpython-3.1.45 idna-3.11 jinja2-3.
1.6 joblib-1.5.2 jsonschema-4.25.1 jsonschema-specifications-2025.9.1 kiwisolver-1.4.9 matplotlib-3.10.7 narwhals-2.9.0 numpy-2.3.4 openpyxl-3.1.5 p
ackaging-25.0 pandas-2.3.3 pillow-11.3.0 protobuf-6.33.0 pyarrow-21.0.0 pydeck-0.9.1 pyogrio-0.11.1 pyparsing-3.2.5 pyproj-3.7.2 python-dateutil-2.9
.0.post0 pytz-2025.2 referencing-0.37.0 requests-2.32.5 rps-py-0.28.0 scikit-learn-1.7.2 scipy-1.16.2 seaborn-0.13.2 shapely-2.1.2 six-1.17.0 smmap
-5.0.2 streamlit-1.50.0 tenacity-9.1.2 threadpoolctl-3.6.0 toml-0.10.2 tornado-6.5.2 typing-extensions-4.15.0 tzdata-2025.2 urllib3-2.5.0 watchdog-6
.0.0 xlswriter-3.2.9

[notice] A new release of pip available: 22.3.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip

(venv) C:\Untar\Skripsi\Tugas Akhir\Testing\Jak-Air-main>
```

9. Jalankan aplikasi Streamlit dengan perintah streamlit run app.py.

```
(venv) C:\Untar\Skripsi\Tugas Akhir\Testing\Jak-Air-main>streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.0.103:8501
```

10. Tunggu beberapa detik, lalu Streamlit akan otomatis membuka browser. Jika tidak, buka secara manual alamat berikut <http://localhost:8501>.



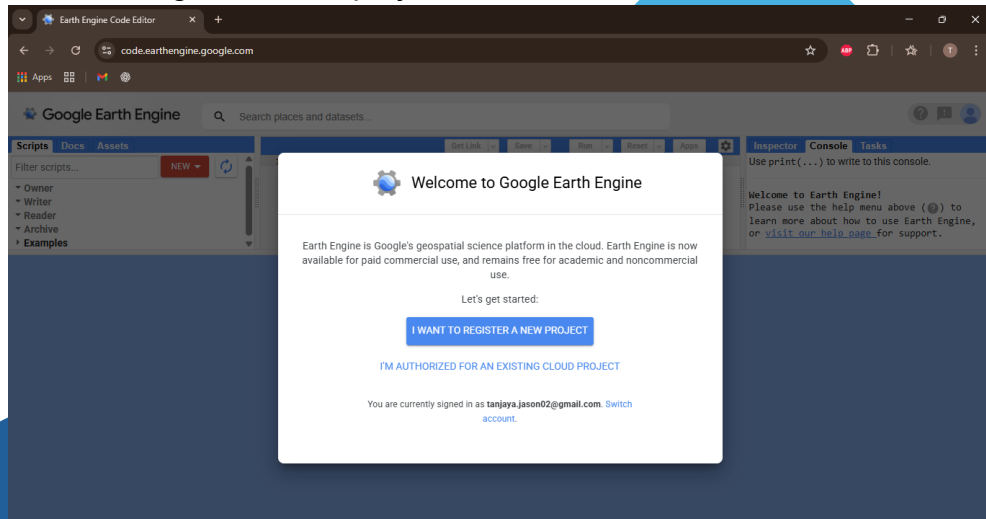
11. Aplikasi siap digunakan secara lokal di browser Anda.
12. Sedangkan untuk menutup atau menghentikan aplikasi, kembali ke terminal tempat aplikasi dijalankan dan tekan 'Ctrl + C'. Kemudian jalankan perintah 'deactivate' untuk keluar dari venv.

```
Stopping...  
(venv) C:\Untar\Skripsi\Tugas Akhir\Testing\Jak-Air-main>deactivate  
C:\Untar\Skripsi\Tugas Akhir\Testing\Jak-Air-main>
```

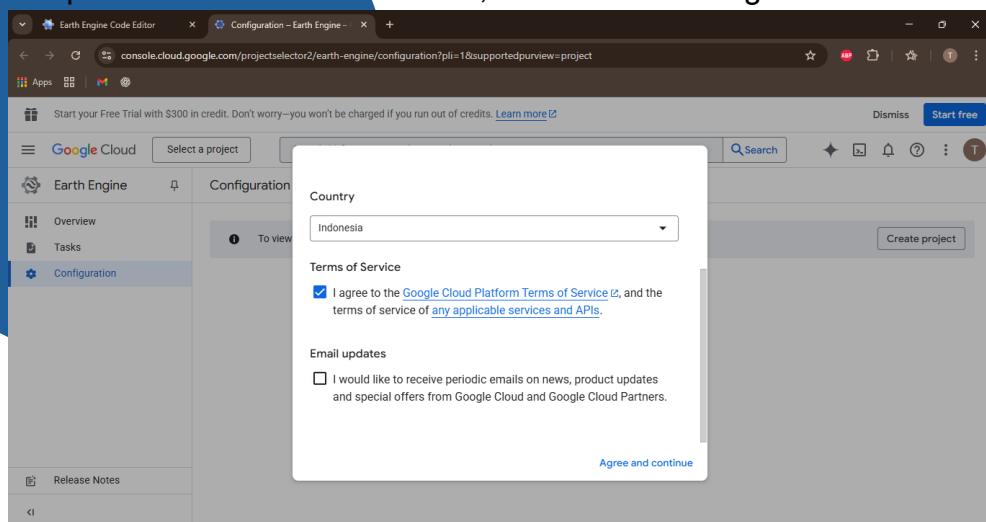
CARA PENGAMBILAN DATA SENTINEL-5P

Data satelit Sentinel-5P tersedia dan dapat diakses secara gratis di platform *Google Earth Engine* melalui situs <https://code.earthengine.google.com/>. Berikut merupakan langkah-langkah pengambilan data satelit Sentinel-5P:

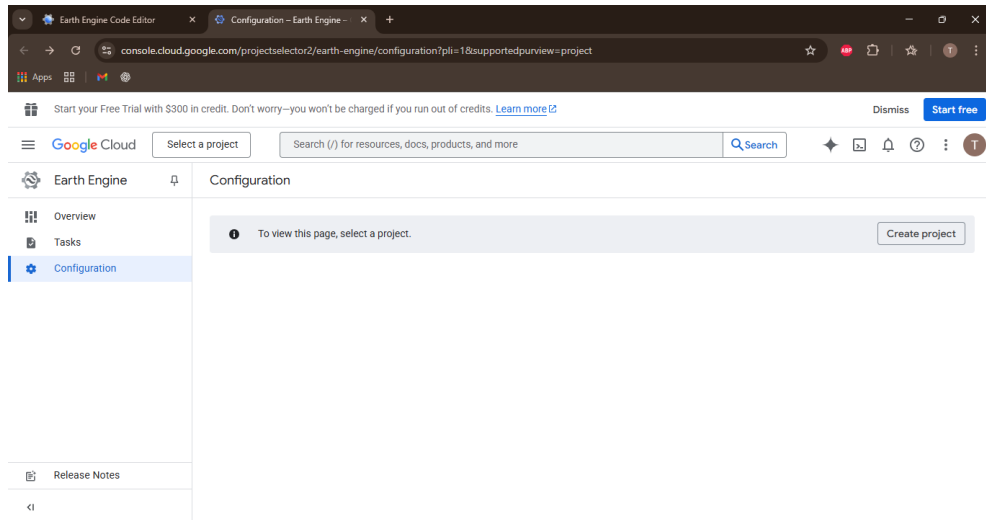
1. Buka situs <https://code.earthengine.google.com/> pada aplikasi web browser dan masuk dengan akun *Google*.
2. Pilih 'I want to register a new project'.



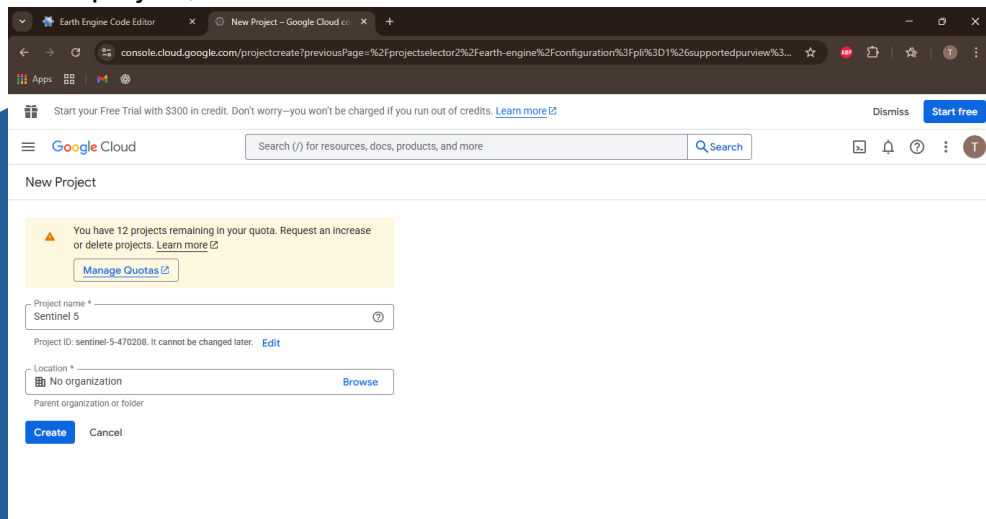
3. Isi ceklis pada kotak 'Terms of service', lalu klik tombol 'Agree and continue'.



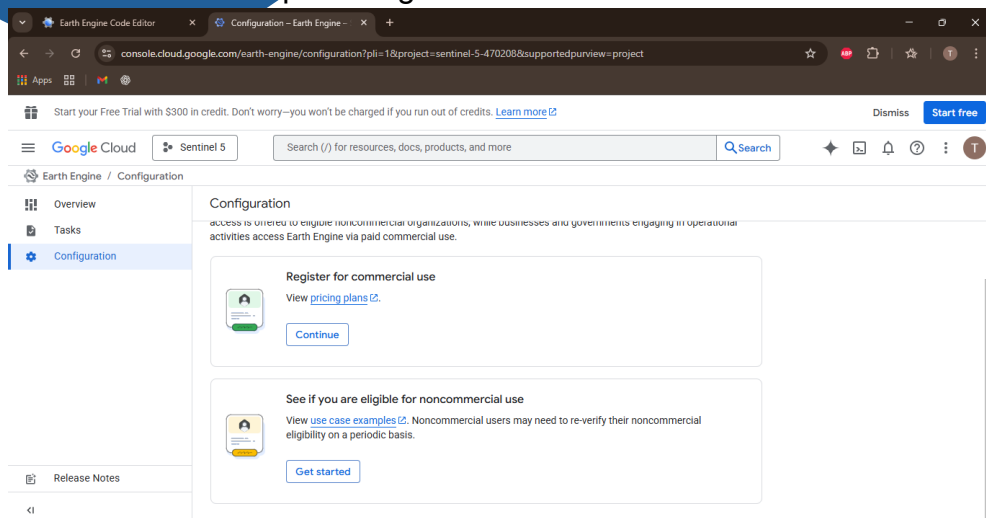
4. Klik tombol 'create project'.



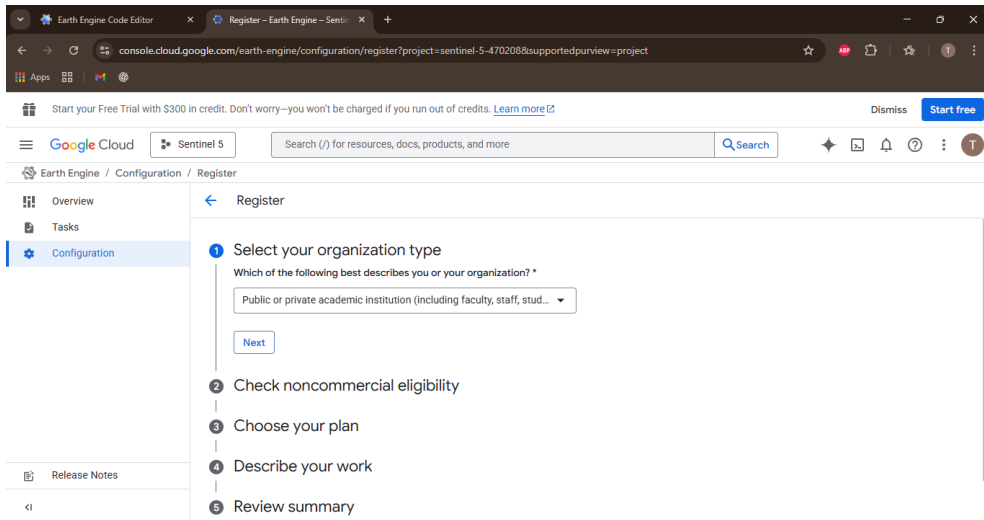
5. Beri nama project, lalu klik tombol 'create'.



6. Klik tombol 'Get started' pada bagian noncommercial use.

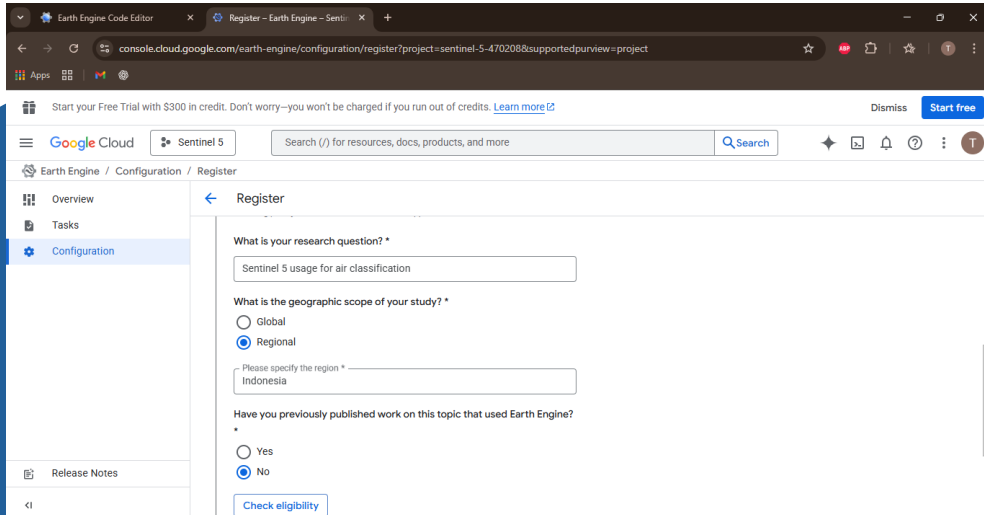


7. Pilih 'Public or private academic institutions'.



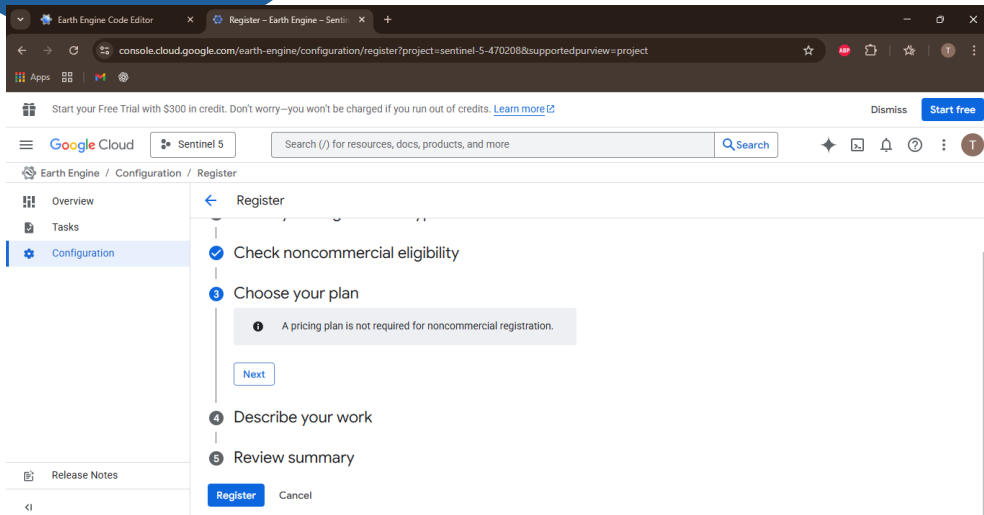
The screenshot shows the 'Register' page in the Google Cloud console for Earth Engine. The left sidebar has 'Configuration' selected. The main content area shows a progress bar with five steps: 1. Select your organization type, 2. Check noncommercial eligibility, 3. Choose your plan, 4. Describe your work, and 5. Review summary. Step 1 is active. The question is 'Which of the following best describes you or your organization? *'. The dropdown menu is open, showing 'Public or private academic institution (including faculty, staff, stud...'. A 'Next' button is below the dropdown.

8. Lengkapi data sesuai kebutuhan.



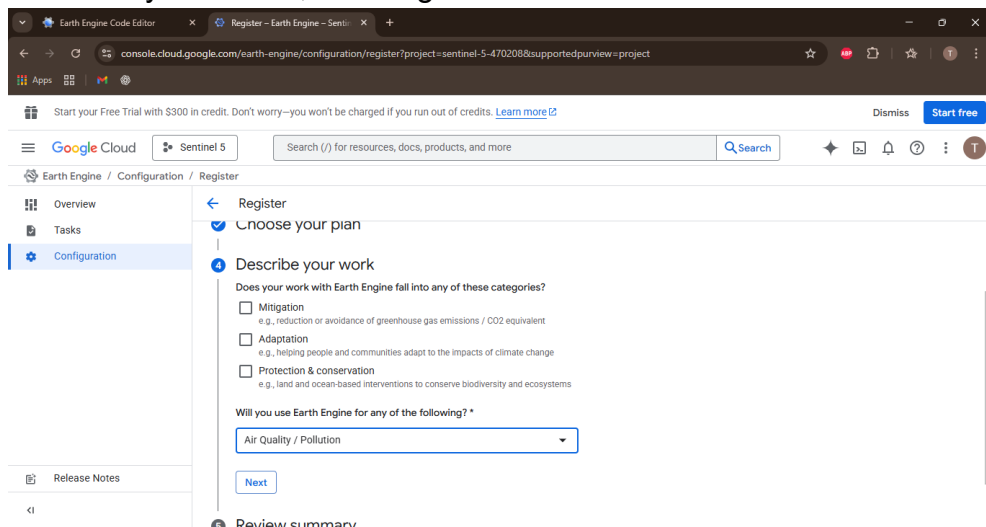
The screenshot shows the 'Register' page in the Google Cloud console for Earth Engine. The left sidebar has 'Configuration' selected. The main content area shows a progress bar with five steps: 1. Select your organization type, 2. Check noncommercial eligibility, 3. Choose your plan, 4. Describe your work, and 5. Review summary. Step 2 is active. The question is 'What is your research question? *'. The text input field contains 'Sentinel 5 usage for air classification'. The question is 'What is the geographic scope of your study? *'. The radio buttons are 'Global' and 'Regional', with 'Regional' selected. The text input field for 'Please specify the region *' contains 'Indonesia'. The question is 'Have you previously published work on this topic that used Earth Engine? *'. The radio buttons are 'Yes' and 'No', with 'No' selected. A 'Check eligibility' button is at the bottom.

9. Pilih next

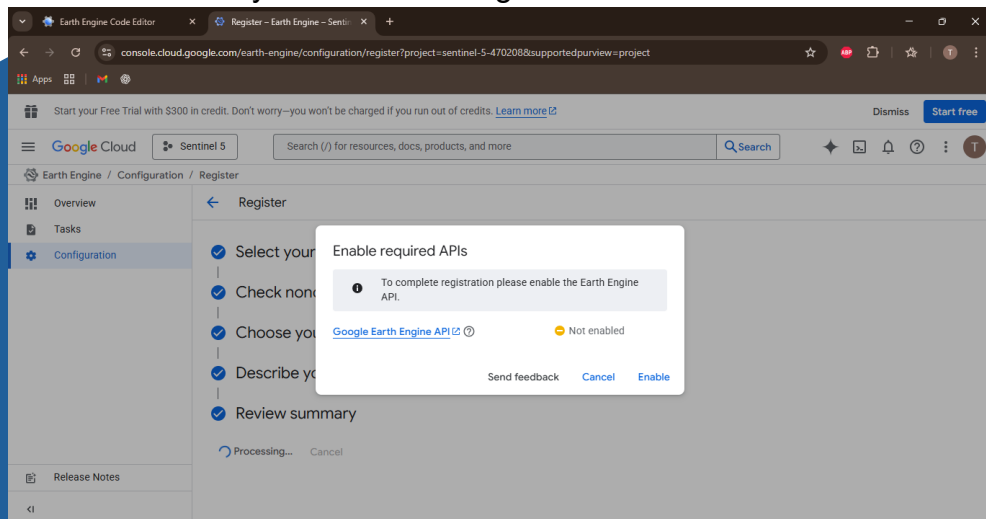


The screenshot shows the 'Register' page in the Google Cloud console for Earth Engine. The left sidebar has 'Configuration' selected. The main content area shows a progress bar with five steps: 1. Select your organization type, 2. Check noncommercial eligibility, 3. Choose your plan, 4. Describe your work, and 5. Review summary. Step 3 is active. The question is 'Choose your plan'. A message box says 'A pricing plan is not required for noncommercial registration.' A 'Next' button is below the message box. The progress bar shows steps 4 and 5 are not yet reached. A 'Register' button and a 'Cancel' button are at the bottom.

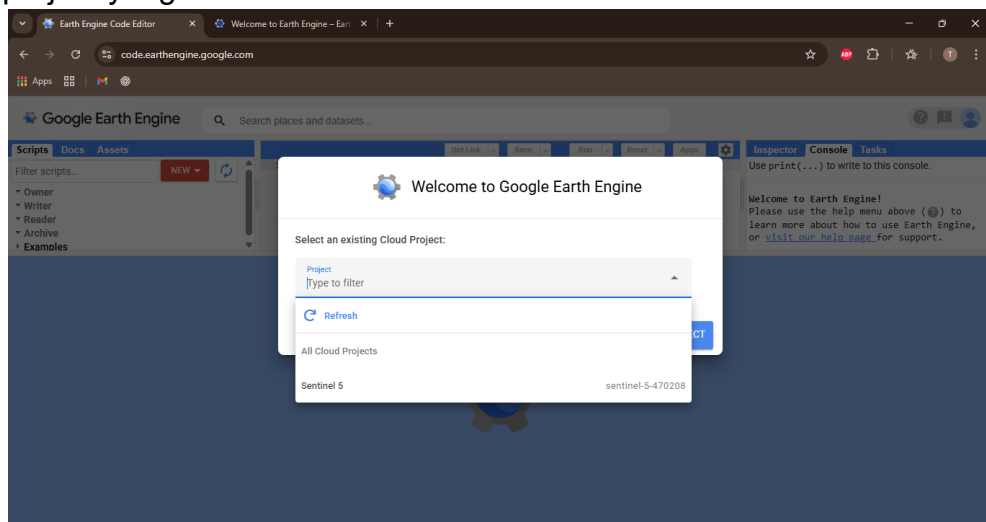
10. Pilih 'Air Quality/Pollution', lalu Register.



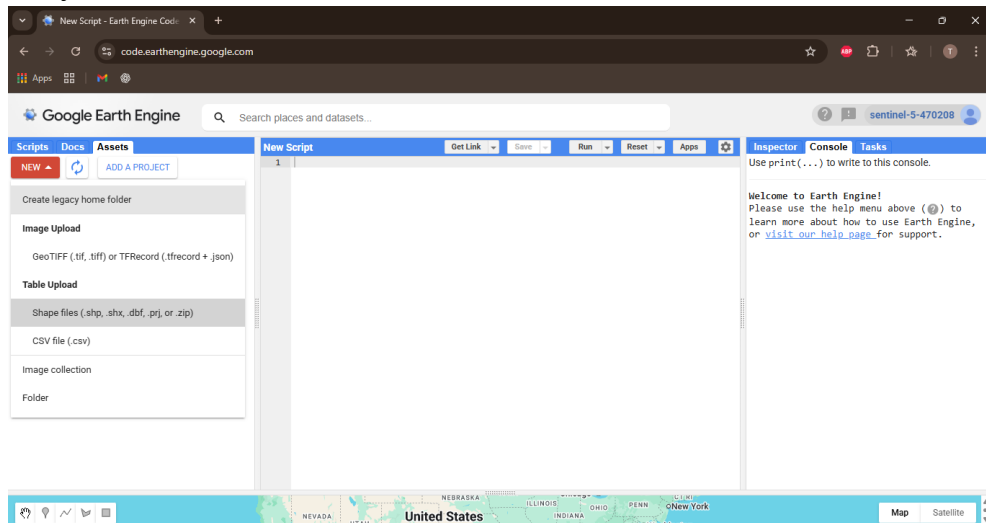
11. Klik enable untuk menyalakan Earth Engine API.



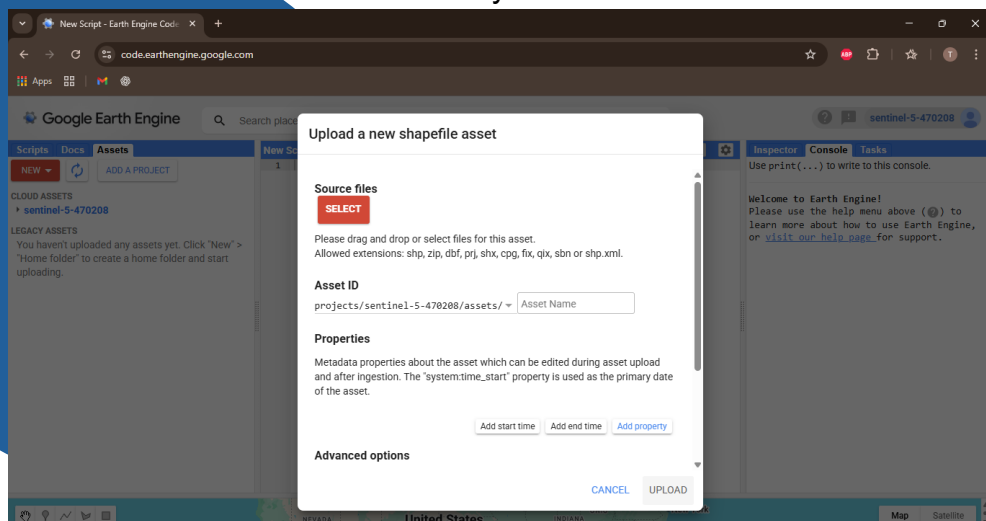
12. Kembali ke halaman Google Earth Engine Code Editor dan refresh. Kemudian pilih project yang tadi dibuat.



13. Klik assets, klik new, dan pilih shape file untuk mengupload shapefile kelima kotamadya.



14. Upload shapefile berupa .dbf, .shp, dan .shx, kemudian beri nama dan pilih upload. Lakukan untuk kelima kotamadya.



15. Isi script berikut untuk mengunduh data satelit Sentinel-5P pada wilayah DKI Jakarta untuk rentang waktu 1 Januari 2025 sampai hari ini.

```
1 // 1. Daftar Aset Shapefile Wilayah Administrasi DKI Jakarta
2 var jakpus_asset = 'projects/sentinel-5-470208/assets/JakPus';
3 var jakut_asset = 'projects/sentinel-5-470208/assets/JakUt';
4 var jakbar_asset = 'projects/sentinel-5-470208/assets/JakBar';
5 var jaksel_asset = 'projects/sentinel-5-470208/assets/JakSel';
6 var jaktim_asset = 'projects/sentinel-5-470208/assets/JakTim';
7
8 // 2. Tentukan Rentang Tanggal untuk Tahun 2025
9 var startDate = ee.Date('2025-01-01');
10 var today = ee.Date(Date.now());
11 var endDate = today; // otomatis sampai hari ini
12
13 // 3. Nama File Ekspor
14 var namaFileEkspor = 'Satelit_Pollutants_DKI_Jakarta_2025_to_' + endDate.format('YYYY-MM-dd').getInfo();
```

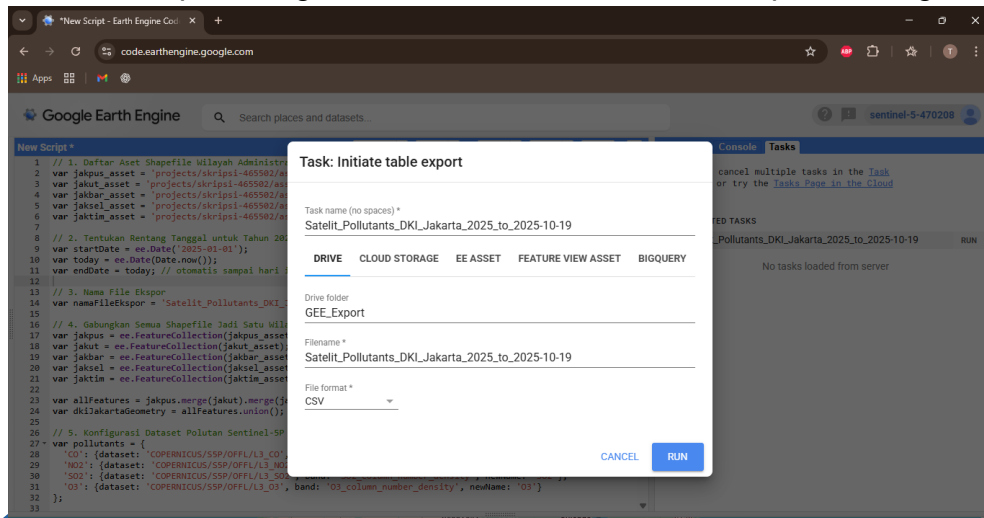
```

16 // 4. Gabungkan Semua Shapefile Jadi Satu Wilayah
17 var jakpus = ee.FeatureCollection(jakpus_asset);
18 var jakut = ee.FeatureCollection(jakut_asset);
19 var jakbar = ee.FeatureCollection(jakbar_asset);
20 var jaksel = ee.FeatureCollection(jaksel_asset);
21 var jaktim = ee.FeatureCollection(jaktim_asset);
22
23 var allFeatures = jakpus.merge(jakut).merge(jakbar).merge(jaksel).merge(jaktim);
24 var dkiJakartaGeometry = allFeatures.union();
25
26 // 5. Konfigurasi Dataset Polutan Sentinel-5P
27 var pollutants = {
28   'CO': {dataset: 'COPERNICUS/S5P/OFFL/L3_CO', band: 'CO_column_number_density', newName: 'CO'},
29   'NO2': {dataset: 'COPERNICUS/S5P/OFFL/L3_NO2', band: 'tropospheric_NO2_column_number_density', newName: 'NO2'},
30   'SO2': {dataset: 'COPERNICUS/S5P/OFFL/L3_SO2', band: 'SO2_column_number_density', newName: 'SO2'},
31   'O3': {dataset: 'COPERNICUS/S5P/OFFL/L3_O3', band: 'O3_column_number_density', newName: 'O3'}
32 };
33
34 // 6. Parameter Spasial
35 var scale = 1113.2;
36
37 // 7. Iterasi Per Hari dalam Rentang 2025
38 var nDays = endDate.difference(startDate, 'day');
39 var dayList = ee.List.sequence(0, nDays.subtract(1));
40
41 var dailyData = dayList.map(function(day) {
42   var currentDate = startDate.advance(ee.Number(day), 'day');
43   var nextDate = currentDate.advance(1, 'day');
44
45   // Fungsi untuk menghitung rata-rata satu polutan (tetap numeric, 4 desimal)
46   var calculateMean = function(pollutantInfo) {
47     var collection = ee.ImageCollection(pollutantInfo.dataset)
48       .filterDate(currentDate, nextDate)
49       .select([pollutantInfo.band], [pollutantInfo.newName]);
50
51     var image = ee.Image(collection.mean());
52
53     return ee.Algorithms.If(
54       collection.size().gt(0),
55       (function() {
56         var value = image.reduceRegion({
57           reducer: ee.Reducer.mean(),
58           geometry: dkiJakartaGeometry,
59           scale: scale,
60           maxPixels: 1e9
61         }).get(pollutantInfo.newName);
62
63         // Jika null, kembalikan null
64         return ee.Algorithms.If(
65           ee.Algorithms.IsEqual(value, null),
66           null,
67           ee.Number(value).multiply(1e4).round().divide(1e4) // bulatkan 4 desimal
68         );
69       })(),
70       null
71     );
72   };
73
74   // Ambil rata-rata masing-masing polutan
75   var coValue = calculateMean(pollutants.CO);
76   var no2Value = calculateMean(pollutants.NO2);
77   var so2Value = calculateMean(pollutants.SO2);
78   var o3Value = calculateMean(pollutants.O3);
79
80   // Buat Feature untuk setiap hari
81   return ee.Feature(null, {
82     'tanggal': currentDate.format('YYYY-MM-dd'),
83     'CO_satelit': coValue,
84     'NO2_satelit': no2Value,
85     'SO2_satelit': so2Value,
86     'O3_satelit': o3Value
87   });
88 });
89
90 // 8. Konversi ke FeatureCollection
91 var finalCollection = ee.FeatureCollection(dailyData)
92   .filter(ee.Filter.notNull(['CO_satelit', 'NO2_satelit', 'SO2_satelit', 'O3_satelit']));
93
94 // 9. Cek dan Ekspor ke Google Drive
95 print('Contoh data harian 2025:', finalCollection.first());
96 print('Jumlah hari dengan data tersedia:', finalCollection.size());
97
98 Export.table.toDrive({
99   collection: finalCollection,
100   description: namaFileEkspor,
101   folder: 'GEE_Export',
102   fileFormat: 'CSV',
103   selectors: ['tanggal', 'CO_satelit', 'NO2_satelit', 'SO2_satelit', 'O3_satelit']
104 });

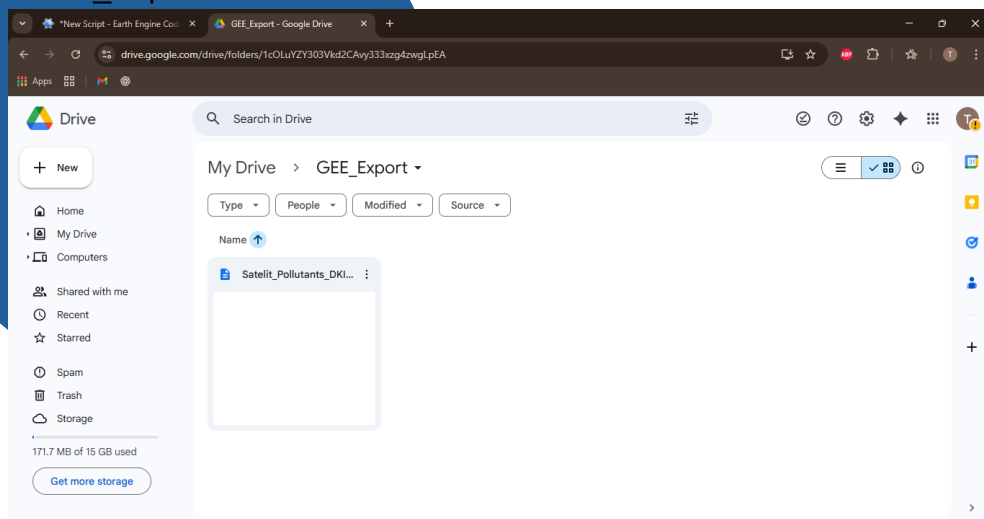
```

16. Klik run, dan pilih tombol 'Tasks'.

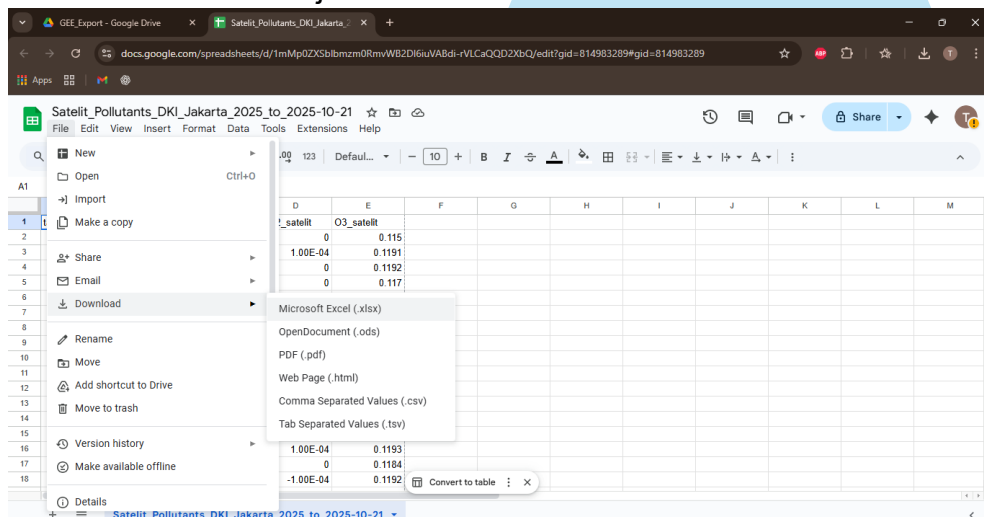
17. Pilih tombol 'run' pada bagian unsubmitted task, kemudian pilih run lagi.



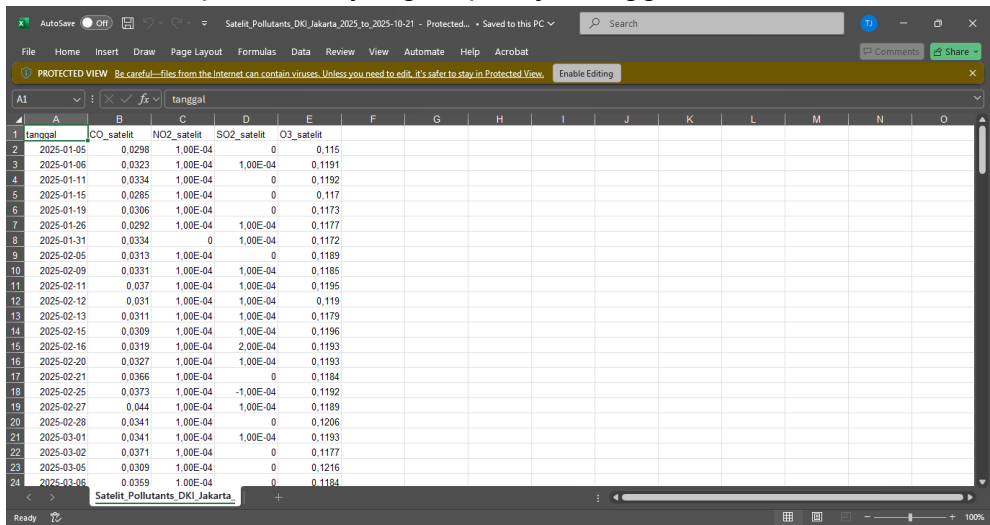
18. Tunggu beberapa saat, setelah task selesai maka data dapat diakses pada drive folder GEE_Export.



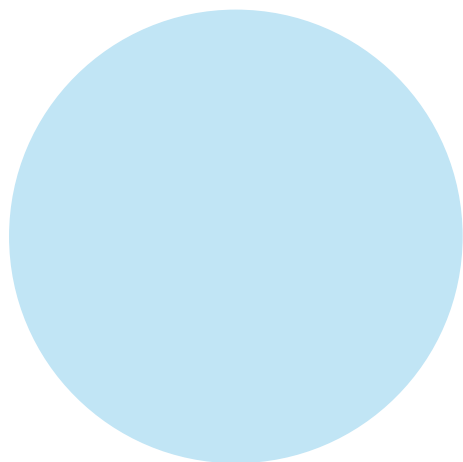
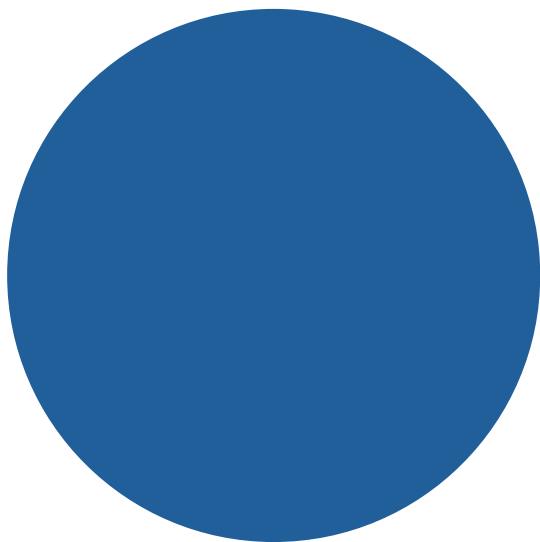
19. Setelah itu, data bisa dibuka menggunakan perangkat lunak Google Sheets. Dari sini data bisa diunduh menjadi format .xlsx.



20. Berikut adalah tampilan data yang siap diuji menggunakan sistem Jak Air.



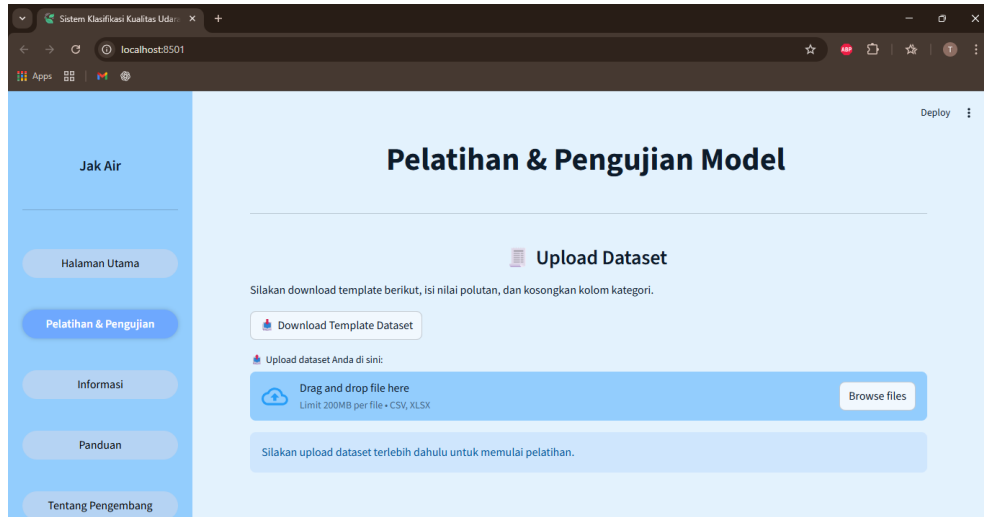
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	tanggal	CO2_satelit	NO2_satelit	SO2_satelit	O3_satelit										
1	2025-01-05	0.0298	1.00E-04	0	0.115										
2	2025-01-06	0.0323	1.00E-04	1.00E-04	0.1191										
3	2025-01-11	0.0334	1.00E-04	0	0.1192										
4	2025-01-15	0.0285	1.00E-04	0	0.117										
5	2025-01-19	0.0306	1.00E-04	0	0.1173										
6	2025-01-26	0.0292	1.00E-04	1.00E-04	0.1177										
7	2025-01-31	0.0334	0	1.00E-04	0.1172										
8	2025-02-05	0.0313	1.00E-04	0	0.1189										
9	2025-02-09	0.0331	1.00E-04	1.00E-04	0.1185										
10	2025-02-11	0.037	1.00E-04	1.00E-04	0.1195										
11	2025-02-12	0.031	1.00E-04	1.00E-04	0.119										
12	2025-02-13	0.0311	1.00E-04	1.00E-04	0.1179										
13	2025-02-15	0.0309	1.00E-04	1.00E-04	0.1196										
14	2025-02-16	0.0319	1.00E-04	2.00E-04	0.1193										
15	2025-02-20	0.0327	1.00E-04	1.00E-04	0.1193										
16	2025-02-21	0.0366	1.00E-04	0	0.1184										
17	2025-02-25	0.0373	1.00E-04	-1.00E-04	0.1192										
18	2025-02-27	0.044	1.00E-04	1.00E-04	0.1189										
19	2025-02-28	0.0341	1.00E-04	0	0.1206										
20	2025-03-01	0.0341	1.00E-04	1.00E-04	0.1193										
21	2025-03-02	0.0371	1.00E-04	0	0.1177										
22	2025-03-05	0.0309	1.00E-04	0	0.1216										
23	2025-03-06	0.0359	1.00E-04	0	0.1184										



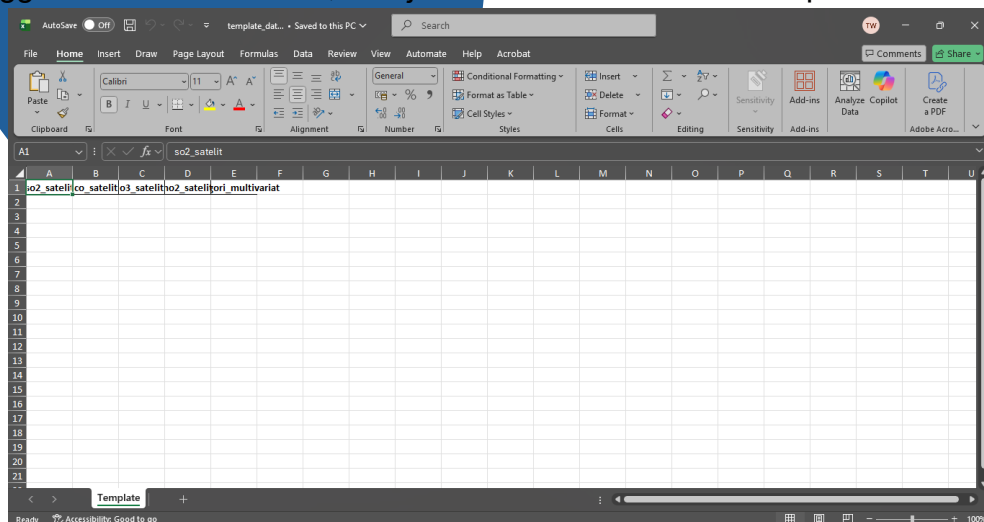
CARA PENGGUNAAN APLIKASI JAK AIR

Berikut merupakan langkah-langkah membuat model klasifikasi kualitas udara pada aplikasi Jak Air:

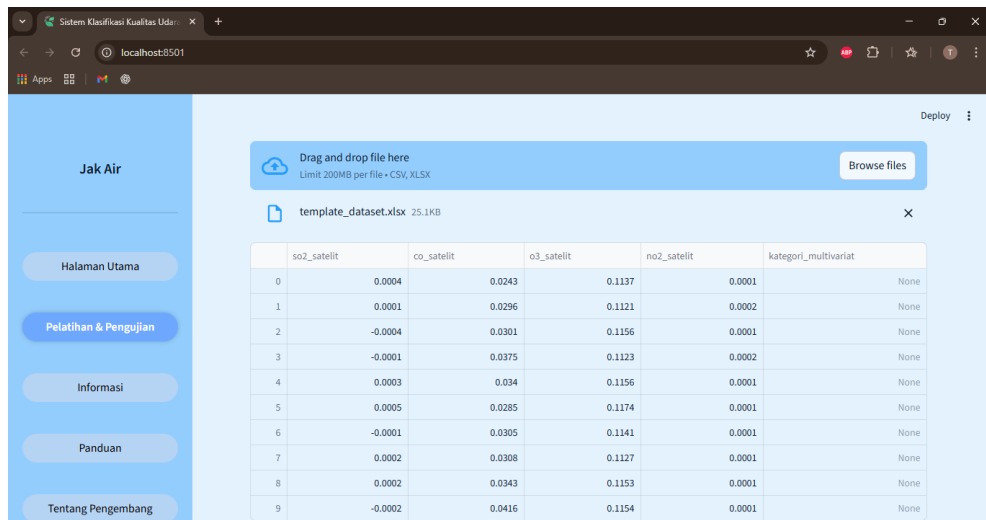
1. Masuk ke halaman Pelatihan dan Pengujian melalui sidebar.



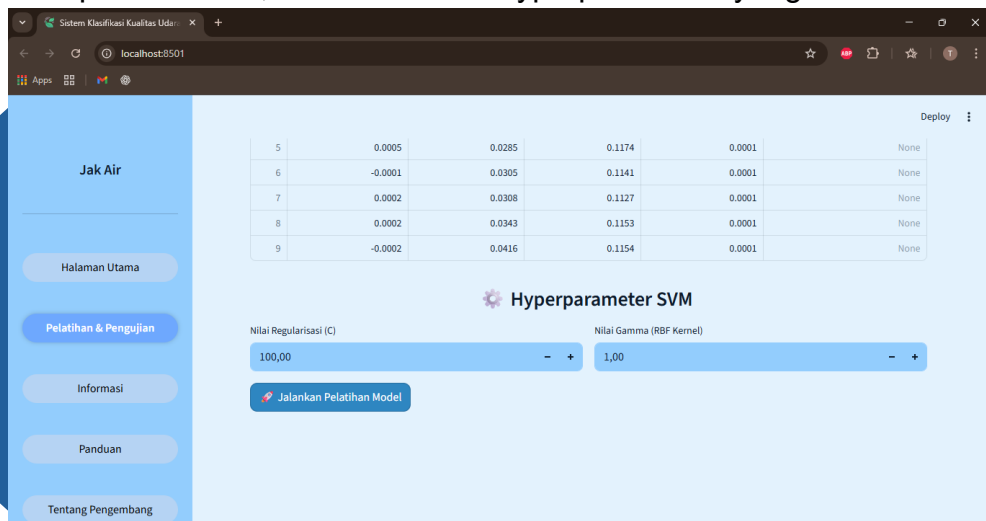
2. Klik tombol Download Template Dataset untuk mengunduh template dataset yang disediakan sistem. Kemudian isikan nilai konsentrasi yang didapat dari data Sentinel-5P pada kolom polutan yang sesuai. Pengguna diharapkan mengisi dataset dengan ketentuan tanpa nilai kosong, tidak mengisi kolom kategori, menggunakan nilai numerik, dan jumlah minimal data 20 sampel.



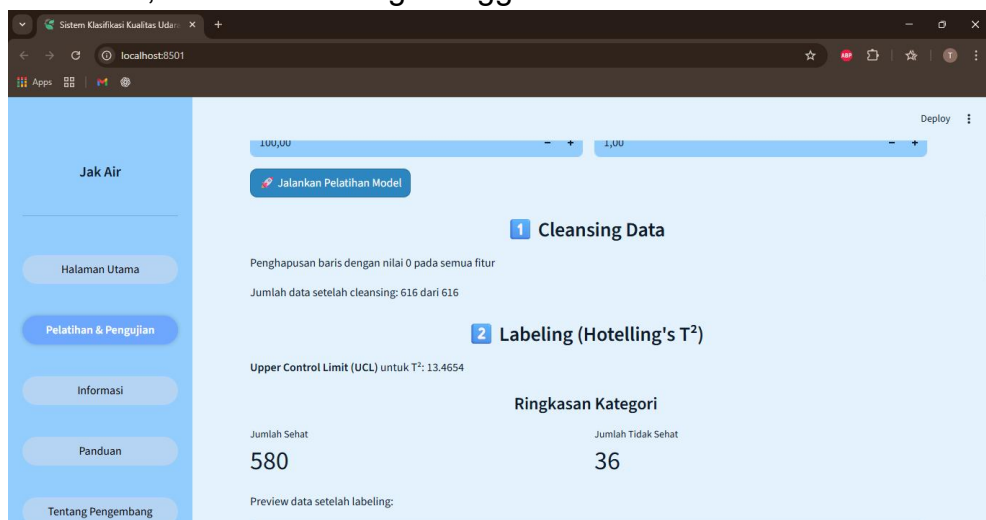
3. Unggah data kembali ke bagian 'Drag and drop file here'. Sistem akan langsung menampilkan preview dari dataset yang diunggah.



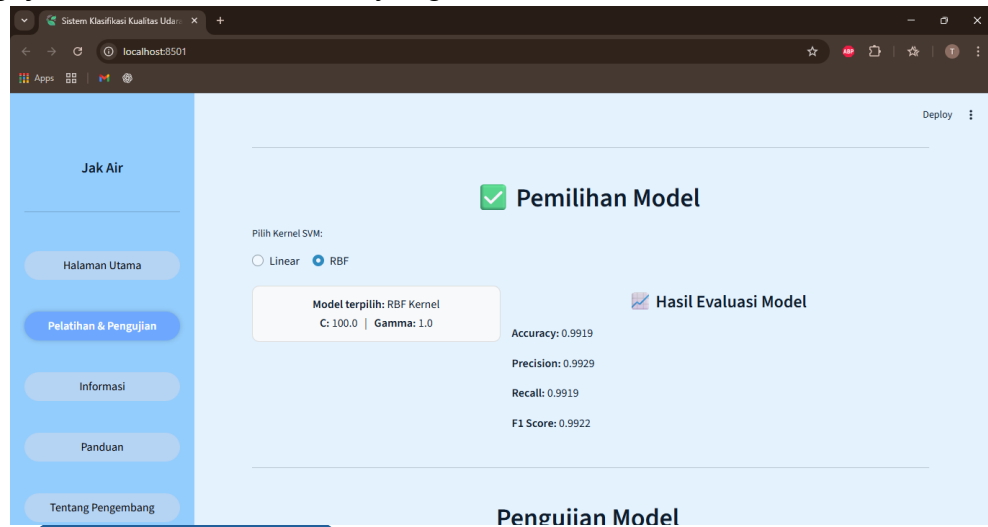
4. Di bawah preview data, masukkan nilai hyperparameter yang dikehendaki.



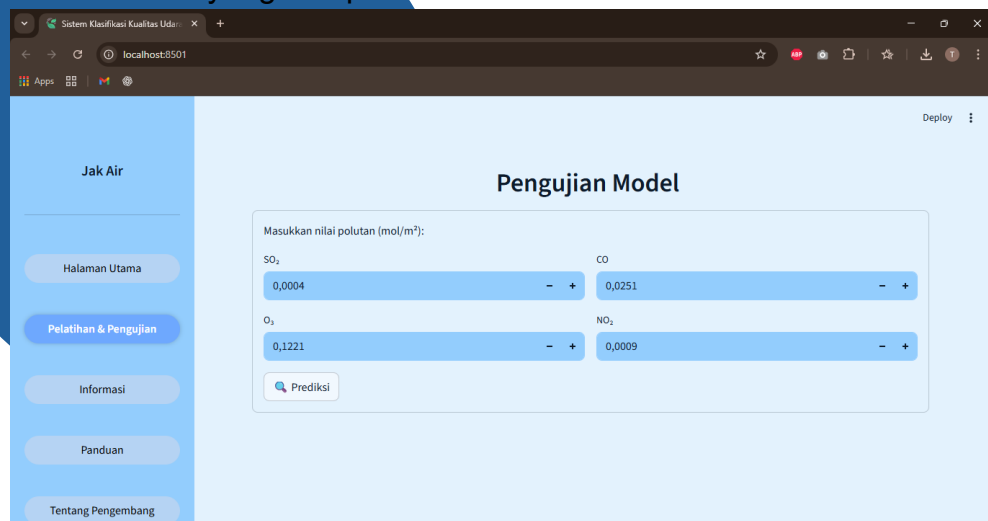
5. Klik tombol Jalankan Pelatihan Model, kemudian tunggu proses sejenak. Sistem akan menampilkan proses yang dijalani oleh data, mulai dari pembersihan data, pelabelan data, dan hasil training menggunakan kedua kernel.



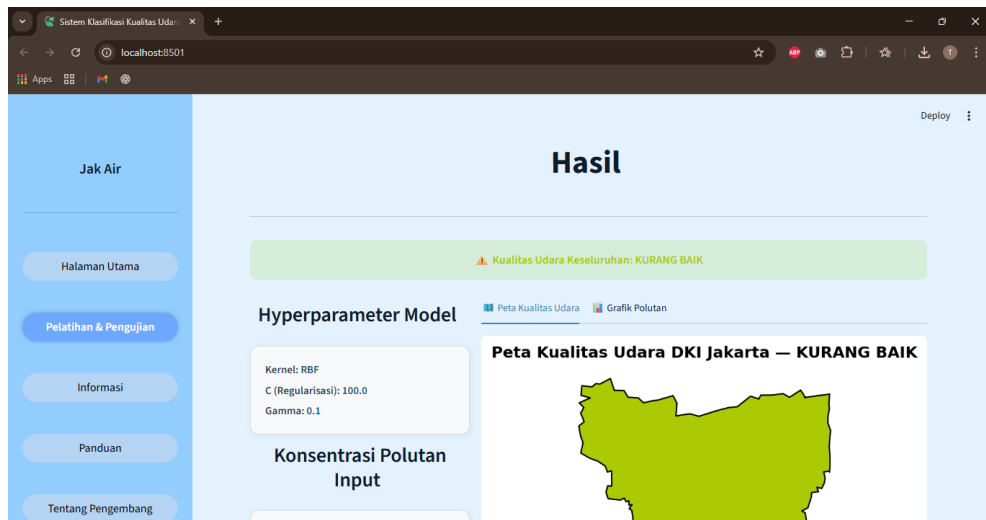
6. Di bawahnya, silakan memilih model mana yang akan digunakan untuk pengujian melalui radio button yang ada.



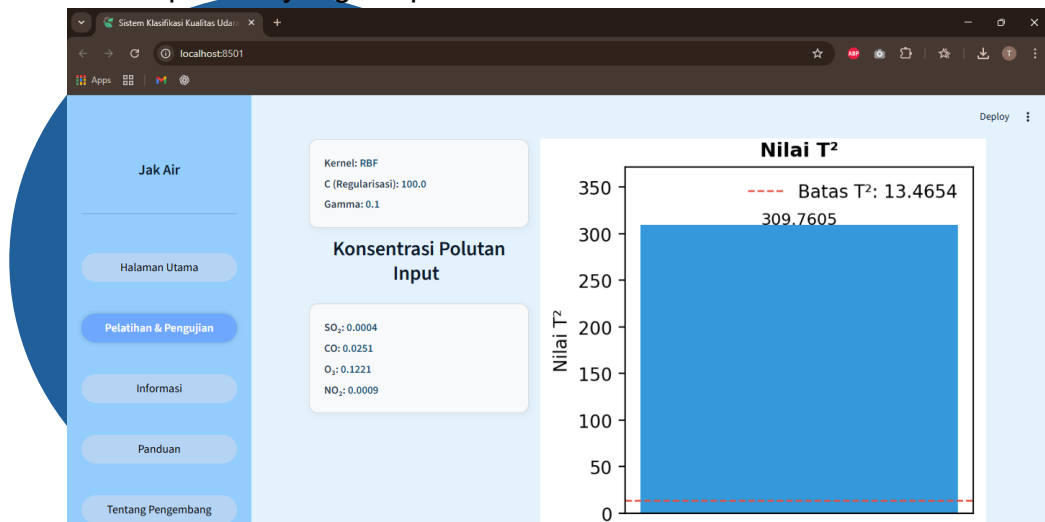
7. Kemudian, masukkan kembali nilai konsentrasi polutan pada kolom pengujian model sesuai data yang didapatkan dari Sentinel-5P.



8. Tekan tombol Prediksi, kemudian tunggu beberapa saat hingga halaman hasil dimuat oleh sistem.
9. Pada halaman Hasil, bisa dilihat hasil klasifikasi kualitas udara keseluruhan dari model yang sudah dibuat, disertai dengan spesifikasi model yang dipakai. Adapun peta yang diberi warna sesuai kategori kualitas udara yang dihasilkan.



10. Bagian peta dapat diganti dengan grafik batang untuk melihat nilai T^2 dari konsentrasi polutan yang diinput



11. Tombol Kembali ke Pengujian bisa digunakan untuk membuat model baru.

