

MODUL PELATIHAN

OTOMATISASI INTERPRETASI PENUTUP LAHAN
UNTUK MENDUKUNG PENYUSUNAN NERACA SUMBERDAYA ALAM DARAT



	Services	Income
12 Norfolk Lane	None	R15,000 p/a
13 Norfolk Lane	Water, Toilet	R11,000 p/a
15 Norfolk Lane	Water, Power	R12,000 p/a
5 Nobo Road	None	



Arc
ESRI GIS



PUSAT PENELITIAN GEOGRAFI TERAPAN

UNIT KERJA KHUSUS LEMBAGA SAINS TERAPAN

FMIPA - UNIVERSITAS INDONESIA

Ged. H Departemen Geografi, Lt. Dasar, FMIPA
Kampus UI, Depok, Jawa Barat, 16425
Telp./Fax : (+62) 21 787-3067 | Web : www.ppgt.ui.ac.id
Email : ppgt@sci.ui.ac.id

DAFTAR ISI

DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	ii
DAFTAR TABEL	iii
BAB 1 DASAR PENGINDERAAN JAUH.....	1
1.1. Konsep Dasar Penginderaan Jauh dan Panjang Gelombang.....	2
1.2. Pengolahan Data Satelit Menggunakan Komputasi Awan	6
BAB 2 PENGENALAN EARTH ENGINE.....	8
2.1. Sign Up	10
2.2. Pengenalan Interface Aplikasi Earthengine.....	11
2.3. Tools Navigasi.....	12
2.4. Mencari Dataset	14
BAB 3 GELOMBANG ELEKTROMAGNETIK	17
3.1. Memunculkan Single Scenes – Greyscale (studi kasus: Landsat 8 Raw)	17
3.2. Memunculkan Single Scenes – RGB (studi kasus: Landsat 8 Raw)	26
3.3. Grafik Z Value (studi kasus: Landsat 8 Raw)	30
BAB 4 PREPROCESSING	40
4.1. Kalibrasi Radiometrik.....	40
4.2. Memotong (Subset) dan mosaic citra (Studi kasus: Sentinel 2A).....	47
BAB 5 PENAJAMAN	60
5.1. Cloud mask (studi kasus: Landsat 8 TOA).....	61
5.2. Pansharpening	64
BAB 6 KLASIFIKASI CITRA.....	69
6.1. Supervised (Studi Kasus Data Landsat 8).....	69

DAFTAR GAMBAR

Gambar 1.1. Jenis sensor berdasarkan sumber energinya	2
Gambar 1.2. Spektrum gelombang elektromagnetik.....	3

Gambar 1.3. Grafik panjang gelombang elektromagnetik	4
Gambar 1.4. Perbedaan resolusi spasial pada citra satelit.....	5
Gambar 1.5. Interpretasi visual citra satelit	7
Gambar 2.1. Tampilan Antarmuka Pengguna Earth Engine.....	12
Gambar 3.1. Spektrometer.....	38

DAFTAR TABEL

Tabel 1.1. Saluran/Kanal pada citra satelit Landsat 8	4
Tabel 6.1. Kedekatan Klasifikasi Penutup Lahan Berdasarkan Skala Peta.....	72
Tabel 6.2. Kelas Tutupan Lahan pada Earth ENgine	72
Tabel 6.3. Kelas Tutupan Lahan Hasil Klasifikasi	80

BAB 1 DASAR PENGINDERAAN JAUH

Penginderaan jauh adalah ilmu dalam mendapatkan dan mengumpulkan informasi mengenai suatu obyek atau fenomena dimuka bumi tanpa menyentuh atau berkontak fisik langsung dengan obyek tersebut. ilmu penginderaan jauh tidak jauh berbeda dengan ilmu fotografi, dimana data yang diolah adalah sama-sama berupa gambar, foto atau data raster. Gambar yang dihasilkan diolah sedemikian rupa, seperti penajaman kontras, pengaturan kecerahan (brightness) dan lainnya sehingga visualisasi citra terlihat lebih indah, informatif dan obyek didalam foto dapat dengan mudah teridentifikasi atau dikenali secara jelas. Oleh karena itu, penginderaan jauh juga terkadang didefinisikan tidak saja sebagai ilmu, namun sebagai suatu seni dalam mengolah dan menafsirkan foto atau gambar untuk mendapatkan suatu informasi. Informasi yang dimaksud tentu adalah informasi obyek, area atau gejala (fenomena) yang terdapat di muka bumi yang bereferensi koordinat.

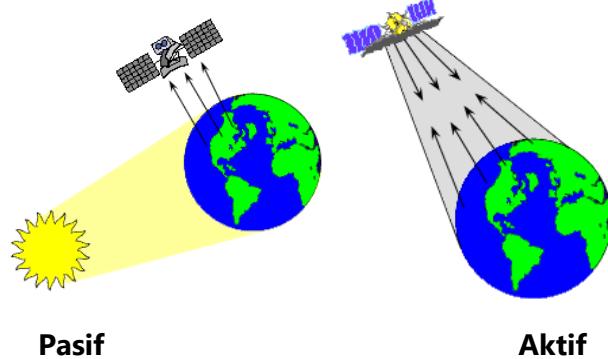
Dalam dunia pemetaan, pemanfaatan teknologi penginderaan jauh ini masuk dalam metode ekstraterrestrial. Metode ekstraterrestrial adalah metode pemetaan yang dilakukan tanpa menyentuh langsung obyek dilapang. Sebagai contoh memetakan sungai akan sangat efektif menggunakan metode ekstraterrestrial dimana drafter atau pembuat peta mendigitasi atau menggambar obyek sungai melalui foto satelit dan tanpa harus turun kelapangan. Kebalikan dari metode ekstraterrestrial adalah metode terrestrial dimana si pembuat peta harus turun kelapang dan menggambar langsung dilapang memanfaatkan alat survey GPS, theodolith, dst.

Pemetaan ekstraterrestrial dengan memanfaatkan data penginderaan jauh memiliki banyak kelebihan dibandingkan pemetaan secara terestrial. Kelebihan tersebut diantaranya:

1. Waktu penggerjaan pemetaan untuk cakupan area yang luas lebih singkat.
1. Tenaga kerja yang dibutuhkan relatif sedikit.
2. Mampu mengidentifikasi area yang sulit untuk dijangkau.
3. Mampu menyajikan peta secara 3 dimensi.
4. Mampu menyajikan kenampakan visual muka bumi secara time series.
5. Biaya lebih murah.

1.1. Konsep Dasar Penginderaan Jauh dan Panjang Gelombang

Prinsip dasar pengambilan data dalam *remote sensing* adalah sensor yang dibawa oleh wahana (satelit, pesawat, pesawat tanpa awak) merekam permukaan bumi dan menghasilkan gambar atau foto. Gambar atau foto yang dihasilkan merupakan hasil dari fenomena interaksi gelombang elektromagnetik terhadap obyek dipermukaan bumi. Sebagai ilustrasi, manusia dapat melihat melalui mata karena terdapat sumber energi dalam bentuk cahaya. Bila tidak terdapat cahaya, maka mata manusia tidak dapat melihat. Mata manusia ini sama halnya dengan sensor yang ada pada satelit. Sensor yang terdapat pada satelit dapat menghasilkan gambar karena terdapat sumber energi berupa cahaya matahari yang memancarkan cahaya ke bumi. Cahaya yang dipancarkan oleh matahari ini merupakan sumber energi yang merupakan manifestasi dari sejumlah spektrum dari gelombang elektromagnetik. Gelombang elektromagnetik yang dipancarkan matahari maka berinteraksi dengan obyek dipermukaan bumi, interaksi tersebut dapat berupa penyerapan, pemantulan dan transmisi gelombang oleh obyek dimuka bumi yang kemudian direkam oleh sensor satelit.



Gambar 1.1. Jenis sensor berdasarkan sumber energinya

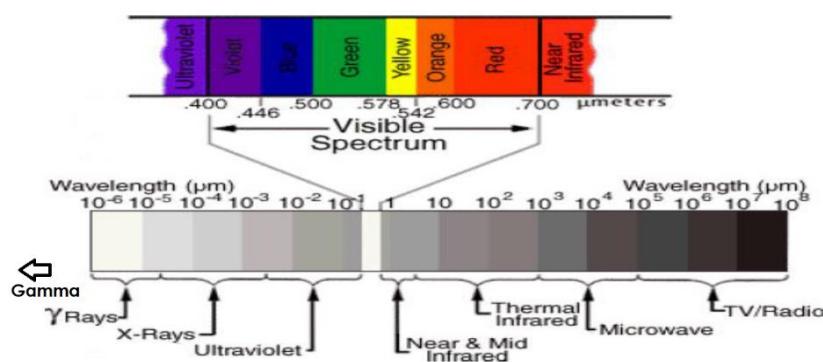
(Sumber: Canadian Center of Remote Sensing, 1986)

Terdapat 2 mekanisme atau tipe penginderaan jauh berdasarkan sumber energi yang digunakan, yakni penginderaan jauh aktif dan penginderaan jauh pasif.

- Penginderaan Jauh Pasif memanfaatkan energi alami seperti matahari atau bulan sebagai sumber energinya. Matahari memancarkan gelombang elektromagnetik ke bumi, lalu gelombang elektromagnetik berinteraksi dengan obyek. Kemudian hasil interaksi tersebut direkam oleh sensor penginderaan jauh dan menghasilkan gambar atau foto atau citra.

- Berbeda dengan sistem pasif, penginderaan jauh aktif menggunakan sensor buatan dalam memancarkan gelombang elektromagnetik, kemudian interaksi gelombang dari muka bumi direkam kembali oleh sensornya.

Dalam skema penginderaan jauh pasif, matahari sebagai sumber energi memancarkan gelombang elektromagnetik dengan spektrum yang kontinyu mulai dari spektrum gamma (Panjang gelombang terkecil) hingga spektrum mikro (Panjang gelombang terbesar). Namun tidak seluruh spektrum tersebut dimanfaatkan dalam penginderaan jauh. Hal ini disebabkan karena tidak seluruh gelombang elektromagnetik dapat mencapai permukaan bumi.

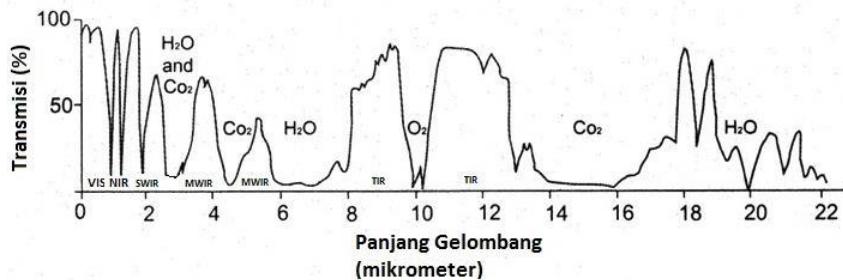


Gambar 1.2. Spektrum gelombang elektromagnetik

Atmosfer yang menyelimuti bumi berfungsi sebagai filter (penyaring) dalam menahan radiasi sinar matahari. Saat gelombang elektromagnetik matahari mencapai permukaan bumi, terdapat 3 interaksi mendasar yang terjadi di atmosfer, yakni:

- Absorpsi, gelombang diserap,
- Transmisi, gelombang elektromagnetik diteruskan, dan
- Refleksi, gelombang dipantulkan.

Ketiga interaksi tersebut terjadi akibat keberadaan berbagai jenis gas yang terdapat di atmosfer seperti oksigen, karbondioksida, nitrogen, hidrogen dan helium. Molekul-molekul gas tersebut dapat menyerap (absorpsi), memantulkan (refleksi) serta meneruskan (transmisi) radiasi elektromagnetik. Gelombang elektromagnetik yang mampu menembus (transmisi mendekati 100%) menuju permukaan bumi itulah yang dapat dimanfaatkan di dalam bidang penginderaan jauh. Fenomena tersaringnya beberapa spektrum gelombang saat melewati atmosfer menuju permukaan bumi ini diistilahkan sebagai jendela atmosfer. Berikut adalah gambaran spektrum yang termasuk kedalam jendela atmosfer beserta gas-gas penghambatnya.



Gambar 1.3. Grafik panjang gelombang elektromagnetik

Setiap satelit penginderaan jauh membawa sensor, sensor tersebut berfungsi untuk merekam data permukaan bumi. Data yang direkam adalah pantulan gelombang elektromagnetik obyek di muka bumi, reflektan tersebut diterima/direkam oleh sensor kemudian dikonversi menjadi suatu gambar atau image. Gambaran muka bumi hasil perekaman satelit inilah yang disebut sebagai data citra satelit.

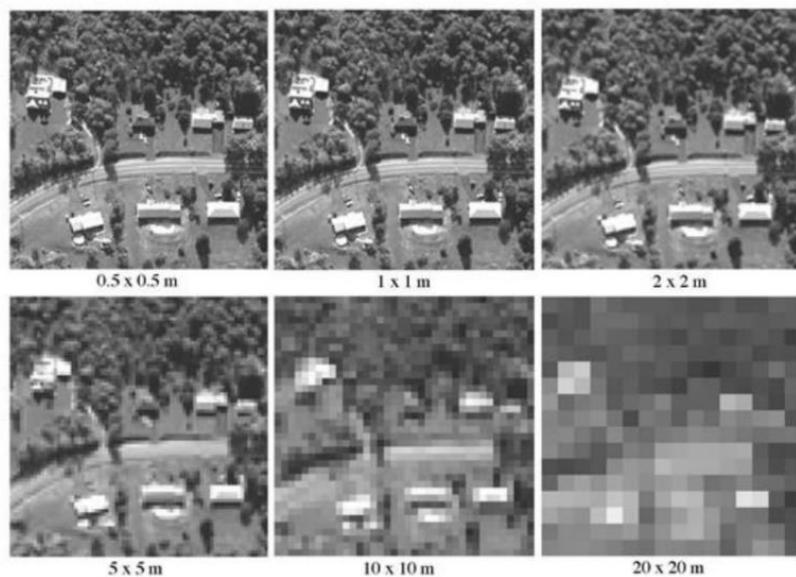
Sensor dalam penginderaan jauh umumnya memiliki banyak saluran atau band yang berfungsi untuk merekam permukaan bumi dalam panjang gelombang yang berbeda-beda. Saluran atau Band pada citra satelit merupakan saluran pada sensor yang menerima gelombang elektromagnetik balik pada panjang gelombang tertentu. Panjang gelombang dimasing-masing saluran band tersebut ditentukan sesuai dengan aplikasi serta misi dari satelit yang membawa sensor tersebut. Selain itu, pemilihan interval panjang gelombang juga harus memperhitungkan jendela atmosfer sehingga gelombang elektromagnetik yang digunakan tidak terhambat oleh atmosfer.

Tabel 1.1. Saluran/Kanal pada citra satelit Landsat 8

Saluran/Band	Aplikasi
Band 1 (0,45 - 0,51 μm)	<ul style="list-style-type: none"> - Tanggap peningkatan penetrasi air. - Mendukung analisis sifat khas lahan, tanah, vegetasi.
Band 2 (0,53 - 0,61 μm)	<ul style="list-style-type: none"> - Mengindera puncak pantulan vegetasi. - Menekankan perbedaan vegetasi dan nilai kesuburan.
Band 3 (0,63 – 0,69 μm)	<ul style="list-style-type: none"> - Memisahkan vegetasi - Klorofil dan kontras vegetasi
Band 4 (0,78 – 0,90 μm)	<ul style="list-style-type: none"> - Tanggap biomass vegetasi - Identifikasi dan kontras tanaman, tanah, air
Band 5 (1,55 – 1,75 μm)	<ul style="list-style-type: none"> - Menentukan jenis vegetasi dan kandungan airnya - Kelembapan tanah
Band 6 (10,4 – 12,5 μm)	<ul style="list-style-type: none"> - Deteksi suhu obyek - Analisis gangguan vegetasi - Kelembapan tanah
Band 7 (2,09 – 2,35 μm)	<ul style="list-style-type: none"> - Pemisahan formasi batuan - Analisis bentuk lahan
Band 8 (0,50 – 0,90 μm)	<ul style="list-style-type: none"> - Pemetaan planimetrik - Identifikasi permukiman - Bentang alam dan budaya - Identifikasi geologi

Dalam penginderaan jauh, ada 4 macam resolusi yang menjadi spesifikasi suatu citra satelit. Keempat resolusi ini haruslah dipahami oleh sang interpreter untuk bisa memutuskan citra apa yang harusnya efektif digunakan.

- Resolusi spasial yaitu ukuran obyek terkecil yang masih dapat diidentifikasi pada citra. Setiap sensor pada satelit memiliki spesifikasi resolusi spasial yang beragam yang menghasilkan tingkat kedetailan foto citra yang berbeda-beda. Foto satelit merupakan data raster atau data gambar yang tersusun dari banyak piksel. Ukuran piksel inilah yang memberikan kedetailan visual dari suatu citra. Suatu obyek dapat teridentifikasi bila ukurannya lebih besar dari ukuran piksel citra satelit. Contoh, sebuah mobil yang ukurannya 2×3 meter, maka mobil tersebut tidak dapat teridentifikasi bila menggunakan Satelit Landsat yang ukuran pikselnya adalah 30×30 meter. Namun apabila menggunakan satelit Pleiades yang memiliki ukuran piksel $0,5 \times 0,5$ m, maka mobil tersebut dapat teridentifikasi. Sehingga dapat disimpulkan bahwa Citra Landsat memiliki resolusi spasial 30 meter dimana lebih rendah bila dibandingkan satelit Pleiades yang memiliki resolusi 0,5 meter.



Gambar 1.4. Perbedaan resolusi spasial pada citra satelit

- Resolusi Temporal yakni periode waktu ulang satelit kembali merekam di area yang sama. Contoh citra satelit yang tergolong memiliki resolusi temporal tinggi, yakni NOAA (4 kali dalam sehari), Modis (1 hari), TRMM (1 hari), dll. Sedangkan citra satelit yang tergolong memiliki resolusi temporal rendah antara lain ALOS (46 hari), Landsat 8 (31 hari), SPOT (26

hari), dll. Biasanya citra satelit yang memiliki resolusi temporal tinggi tersebut adalah satelit yang digunakan untuk mengidentifikasi kondisi atmosfer dan cuaca. Hal tersebut tentu dimaksudkan karena kondisi atmosfer dan cuaca sangat dinamis perubahannya. Satelit yang memiliki resolusi temporal tinggi umumnya memiliki resolusi spasial yang sangat rendah.

- Resolusi Radiometrik adalah ukuran sensitivitas sensor dalam membedakan spektral maupun radiasi suatu obyek. Contoh, citra Landsat 7 memiliki sensor dengan resolusi radiometrik 8 bit, artinya sensor tersebut dapat membedakan suatu obyek sebanyak $2^8 = 256$ variasi. Sehingga nilai piksel pada Landsat 7 memiliki interval 0-255. Sedangkan generasi Landsat selanjutnya yakni Landsat 8 memiliki resolusi radiometrik 16 bit, artinya sensor Landsat 8 lebih sensitif dalam membedakan obyek sebanyak $2^{16} = 65536$ variasi dibandingkan Landsat 7.
- Resolusi Spektral yakni banyaknya jumlah saluran/band spektral yang digunakan pada citra. dalam resolusi spektral ini, citra satelit digolongkan menjadi 2 jenis, yakni citra multispektral dan citra hyperspektral. Citra multispektral adalah citra satelit yang memiliki jumlah saluran/band kurang dari 30. Contoh citra multispektral seperti Landsat 8 (11 band), SPOT 4 dan 5 (4 band), ALOS AVNIR (4 band), dll. Sedangkan citra hyperspektral adalah satelit yang memiliki jumlah band > 30 pada sensornya, contohnya adalah citra Modis dan citra Hyperion.

1.2. Pengolahan Data Satelit Menggunakan Komputasi Awan

Data satelit penginderaan jauh dapat kita peroleh baik secara gratis maupun yang berbayar. Umumnya, data satelit berbayar ini memiliki karakteristik resolusi spasial yang sangat tinggi yakni 5 meter bahkan hingga dibawah 0,5 meter. Sedangkan data satelit yang tersedia secara gratis umumnya memiliki resolusi rendah hingga menengah. Resolusi spasial tertinggi untuk kategori data satelit yang *free* adalah sentinel 2A yang memiliki resolusi 10 meter. Adapun citra satelit resolusi sangat tinggi umumnya digunakan untuk pemetaan skala detail atau pemetaan kadaster, dimana umumnya pengolahannya lebih banyak menerapkan interpretasi secara visual dibandingkan digital. Walaupun banyak teknik klasifikasi digital untuk data satelit resolusi tinggi, namun akurasinya tidak sebaik dibandingkan dengan klasifikasi manual (visual interpretation). Sedangkan data satelit resolusi menengah umumnya digunakan

untuk pemetaan wilayah dengan cakupan yang luas, sehingga sangat efektif bila diolah menggunakan pengolahan digital.



Gambar 1.5. Interpretasi visual citra satelit

Data satelit merupakan kategori data raster, dimana semakin tinggi resolusi spasialnya serta semakin besar cakupan area yang diolah maka semakin besar ukuran datanya. Ukuran data adalah masalah utama dalam mengolah data citra satelit, sebab:

1. Mendownload data dari penyedia data membutuhkan kuota internet.

BAB 2 PENGENALAN EARTH ENGINE

Google Earth Engine merupakan platform baru yang memungkinkan kita untuk mengamati dan memonitor kondisi lingkungan bumi. Platform ini menggunakan konsep pengolahan data berbasis komputasi awan yang didukung ribuan mesin komputasi performa sangat tinggi yang berada di Google data center sehingga memungkinkan kita untuk melakukan analisis data yang besar (Big Data Analysis). Karena menggunakan komputasi awan, maka proses penggunaanya dilakukan secara online dan membutuhkan koneksi internet.

Fitur unggulan yang terdapat pada earthengine adalah:

a. Dataset

Earthengine dapat terhubung dengan database berbagai macam jenis data satelit penginderaan jauh. Anda dapat mengolah data satelit tanpa harus mendownload data mentahnya (raw data), cukup mengimport dataset dari beberapa penyedia data seperti USGS, ESA, JAXA, dll. Raw data tersebut diolah sedemikian rupa sesuai keperluan analisa didalam server cloud, dan hasilnya dapat anda unduh (download).

b. Komputasi Super

Proses pengolahan data dapat dilakukan dengan sangat cepat. Sebagai contoh, mengolah data dengan ukuran yang besar yang umumnya dapat diselesaikan dalam waktu seminggu bila menggunakan cara konvensional (dekstop), namun menggunakan Aplikasi Earthengine (cloud) ini bisa diselesaikan dalam hitungan jam. Hal ini memungkinkan user tidak membutuhkan spesifikasi hardware tinggi, cukup berbekal koneksi internet saja.

c. Efektivitas Pengolahan

Proses pengolahannya sangat simpel yakni proses pengolahan data yang terdiri dari banyak langkah dan tahapan dapat terselesaikan secara sekaligus dalam satu kali proses saja. Bahkan anda pun tidak perlu menunggu lama untuk memperbaiki hasil bila anda melakukan revisi atau editing dalam hal metodologi dan perubahan sampel data. Kelebihan komputasi awan dibandingkan versi dekstop ini adalah sebanyak apapun kita mengulang proses, tidak akan memenuhi hardisk kita karena output tersimpan didalam server. Bila hasil analisis sudah dianggap benar, barulah anda dapat mengunduhnya melalui link googledrive.

Adapun dari sisi kekurangan atau kelemahan dari earthengine, diuraikan sebagai berikut:

a. Code Editor Tidak User Friendly

Proses pengolahan data membutuhkan keahlian dan kemampuan Bahasa Javascript. Hal ini dikarena semua rangkain proses, dari mulai input (import) dataset, pengolahan, visualisasi hingga export (download) dilakukan dengan menyisipkan perintah dalam bentuk bahasa javascript di kolom Code Editor. Hal ini tentu membuat user yang umumnya berlatar belakang keilmuan penginderaan jauh, harus mempelajari konsep Bahasa javascript.

b. Fungsionalitas tidak lengkap

Tidak semua proses dan *tools* yang ada didalam Earthengine lengkap. Sebagai contoh, dalam pengolahan data satelit aktif (Radar) yang bersifat agak tingkat lanjut (advance), tidak ditemukan script yang berfungsi untuk melakukan koreksi radiometrik topografi dan Interferometri. Namun, untuk proses atau *tools* yang sifatnya mendasar sudah dikatakan cukup lengkap.

c. Terbatas Penyimpanan

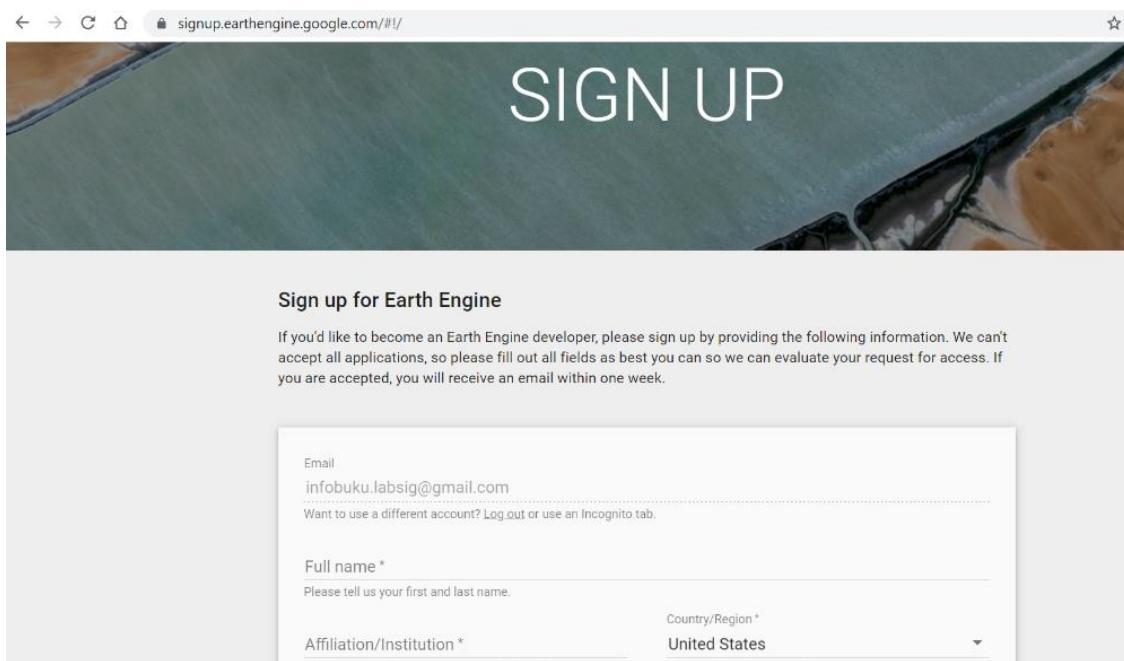
Sebagaimana diketahui bahwa seluruh output hasil analisis yang dilakukan dapat tersimpan di server cloud. Namun anda dapat mengunduh data output tersebut, atau bahkan anda juga bisa mengunduh raw data kedalam laptop atau pc. Proses mendownload adalah dengan memberikan perintah melalui script terhadap server untuk mengirimkan data yang kita inginkan kedalam akun google drive. Sebagaimana diketahui, setiap akun google drive (free) memiliki keterbatas maksimum 15 GB. Sehingga user harus cermat dengan mengosongkan kapasitas akun google drive supaya bisa berulangkali mendownload file dari earthengine.

Earthengine ini dapat dikatakan memiliki segudang kebermanfatan. Dengan aplikasi Earthengine ini memungkinkan para pakar dan peneliti untuk memantau dan monitoring lingkungan bumi dengan sangat mudah. Aplikasi ini juga dapat dikatakan pioneer dalam memulai era digital 4.0 dalam bidang penginderaan jauh, dimana kedepannya nanti mungkin sudah tidak dibutuhkan lagi *software* berbasis desktop maupun hardware berspesifikasi tinggi sehingga pengolahan tidak lagi dilakukan secara desktop namun menggunakan komputasi awan.

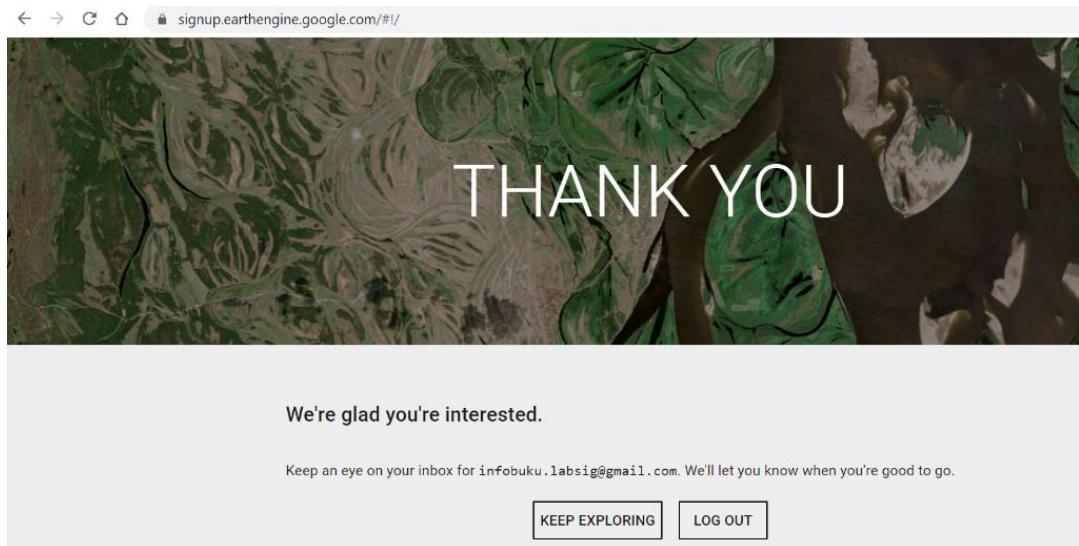
2.1. Sign Up

Sebagaimana pada aplikasi-aplikasi cloud lainnya, aplikasi earthengine dapat berjalan bila anda telah terdaftar dan memiliki akun. Akun yang didaftarkan adalah akun email google, karena earthengine ini merupakan bagian dari produk google. Sebagai saran, buatlah 1 akun email google (gmail) baru dimana akun ini khusus digunakan sebagai akun earthengine. Mengapa? Saat anda selesai mengolah data citra melalui earthengine, maka output datanya bisa anda download melalui google drive akun gmail anda. Umumnya, akun free gmail hanya menyediakan kapasitas 15 GB dan bila akun gmail anda penuh, maka tentu akan sedikit merepotkan dengan menghapus file-file lama anda agar data hasil olahan anda bisa tersimpan di gdrive.

Berikut adalah cara mendaftarkan akun gmail didalam aplikasi earthengine. Masuklah pada website: [signup.earthengine.google.com/#!/](https://signup.earthengine.google.com/#/)



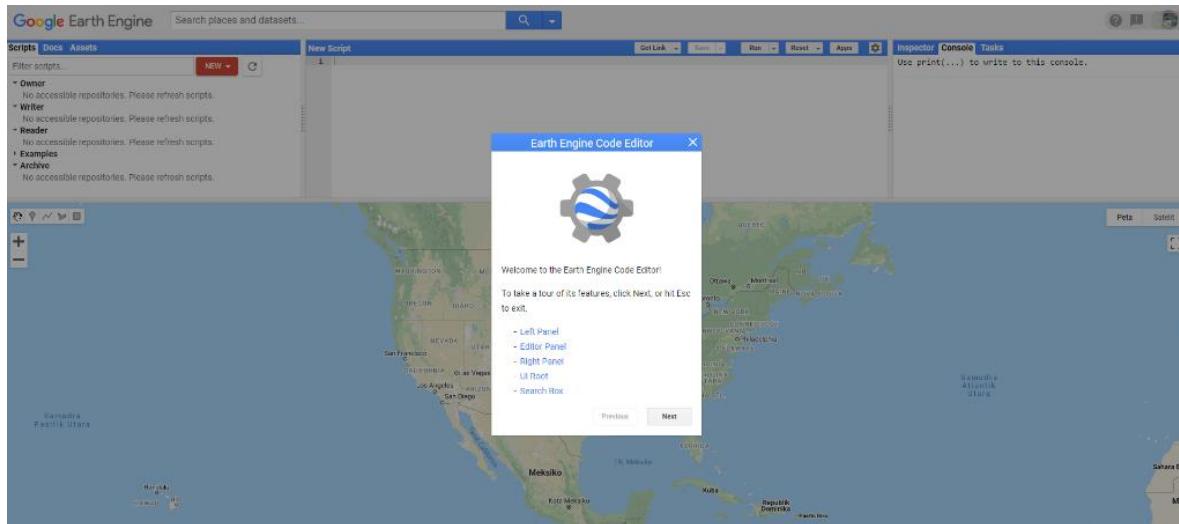
Isilah seluruh informasi pada lembar *sign up* tersebut serta menceklis kolom *agreement term of service*. Bila tampilannya menjadi seperti dibawa ini, maka akun email anda sudah aktif dan terdaftar.



Selanjutnya adalah dapat langsung membuka aplikasi earthengine dengan mengakses link berikut:

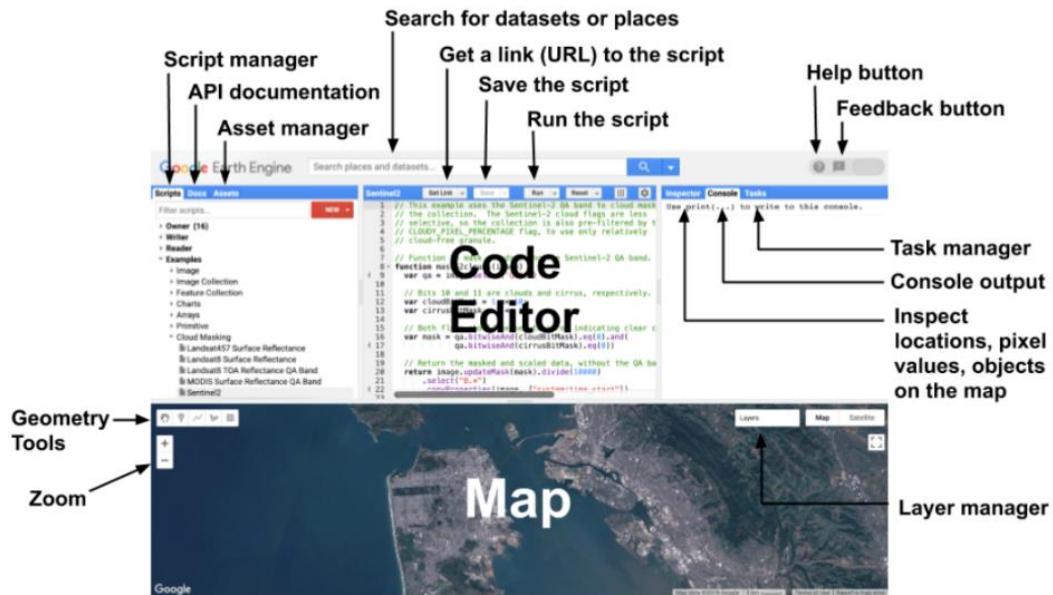
<https://code.earthengine.google.com/#>

gunakan email yang sudah anda aktifkan untuk mengakses link di atas, hingga tampilannya menjadi seperti di bawah ini.



2.2. Pengenalan Interface Aplikasi Earthengine

Dari sisi interface, aplikasi earthengine terdiri dari sisa atas yang berupa 3 kolom dan sisi bawah berupa peta.

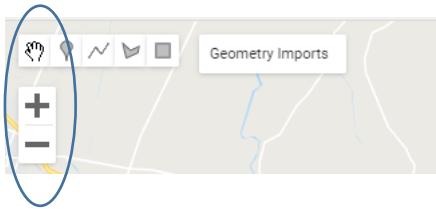


Gambar 2.1. Tampilan Antarmuka Pengguna Earth Engine

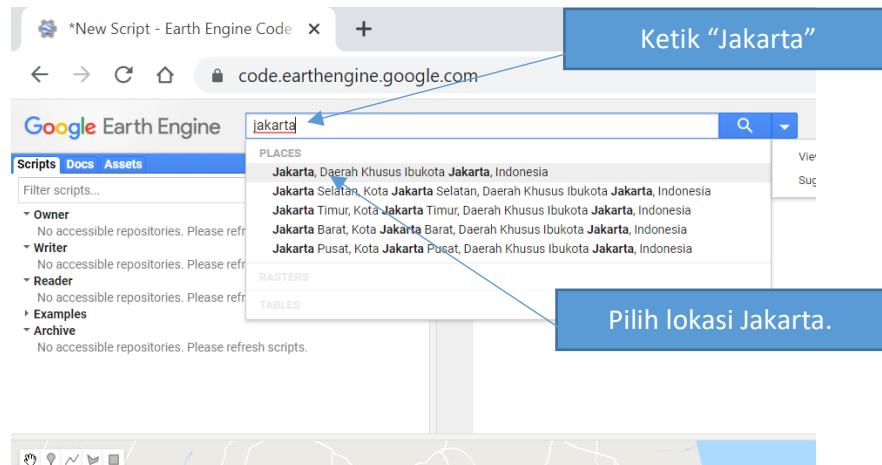
- *Script manager* → tempat penyimpanan script yang telah dibuat
- *Asset manager* → tempat untuk upload data, termasuk mengupload shapefile untuk keperluan pemotongan citra, sampel training data, dll.
- Kolom *search dataset or places* → anda dapat mencari dataset dari berbagai macam jenis satelit atau informasi spasial lainnya. Selain itu, anda dapat juga mencari suatu lokasi melalui kolom ini.
- *Save the script* → tombol untuk menyimpan script yang telah anda buat
- *Run the script* → untuk menjalankan atau memulai proses dari script atau perint
- *Inspector* → memberikan informasi nilai piksel dan koordinat lokasi di peta.
- *Console* → tempat munculkan informasi-non spasial terkait perintah dari script yang anda buat, misalkan list data, table, grafik, dll.
- *Task* → rincian dari seluruh tugas yang akan atau telah selesai.
- *Code Editor* → kolom untuk membuat perintah melalui penulisan script.

2.3. Tools Navigasi

Anda dapat mengaktifkan tombol untuk menggeser peta atau tombol + dan - untuk memperbesar dan memperkecil peta. Bila anda menggunakan mouse, anda dapat menggunakan scroll pada mouse untuk memperkecil dan memperbesar tampilan peta.

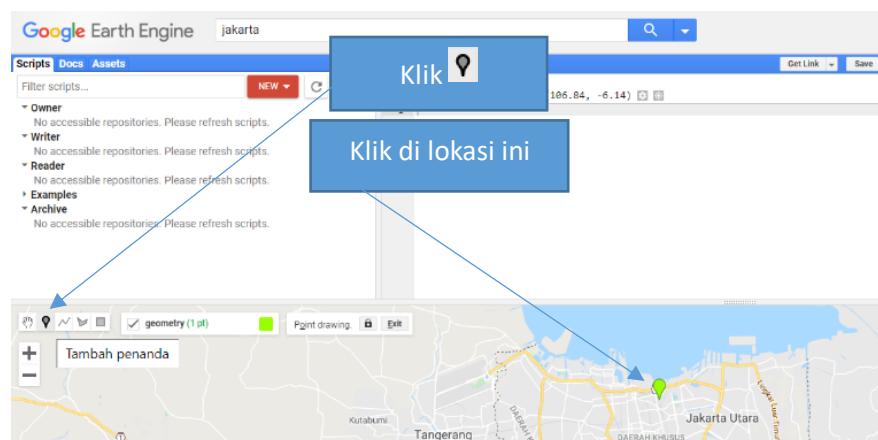


Untuk mencari lokasi, anda dapat mencari melalui kolom *search*, sebagai contoh ketiklah **Jakarta**, maka peta akan menunjukkan lokasi Jakarta.



Selanjutnya anda bisa menggunakan penanda, misalkan titik atau menggambar polygon. Penanda ini digunakan umumnya untuk mencari dataset pada suatu lokasi yang diinginkan. Maka, lokasi tersebut cukup anda plot bisa dalam bentuk titik, atau bentuk polygon.

Aktifkan tombol , lalu mark atau plot di peta hingga muncul penanda berbentuk balon bewarna pada peta.



Selain muncul penanda lokasi berbentuk balon, di dalam kolom code editor juga muncul script dari penanda lokasi yang sudah dibuat.

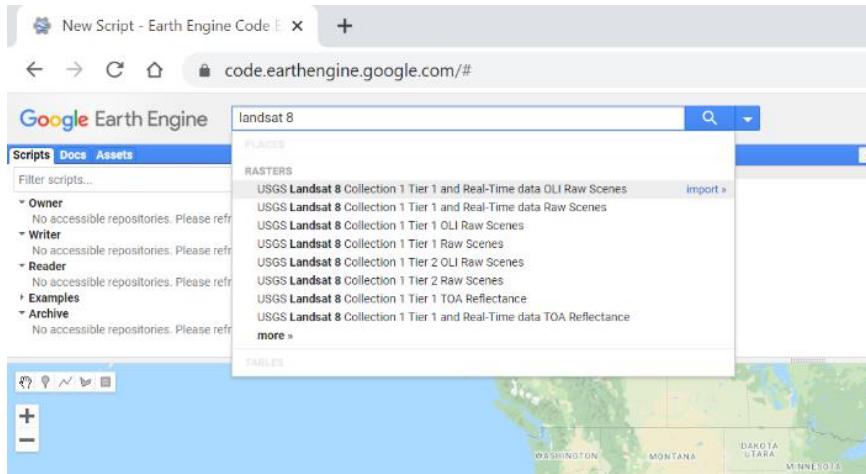
```
New Script * Get
Imports (1 entry) ↗
var geometry: Point (106.84, -6.14) ⚙️ ⓘ
1 |
```

Dalam konteks ini, anda sudah membuat sutau variable baru, yakni variable lokasi, dengan nama variable "geometry" sebagai default-nya. Sekarang coba ganti nama (*rename*), misalkan menjadi "Jakarta".

```
New Script * Get
Imports (1 entry) ↗
var jakarta: Point (106.84, -6.14) ⚙️ ⓘ
1 |
```

2.4. Mencari Dataset

Selanjutnya adalah mencari dataset. Didalam earthengine, terdapat banyak dataset dari berbagai jenis citra satelit. Anda dapat mencarinya melalui kotak search, misalkan ketiklah "Landsat 8", untuk mencari citra satelit landsat 8.



Untuk kategori citra satelit Landsat 8 pun tersedia beragam, sebagai contoh terdapat Landsat 8 tipe data RAW, terdapat juga Data Landsat 8 yang sudah terkalibrasi radiometrik, dll. Sebagai contoh, pilihlah *Landsat 8 Collection 1 Tier 1 and Real-Time data Raw Scenes*. Maka akan muncul box berisi informasi dari data Landsat 8 jenis tersebut. Anda dapat melihat Data availability-nya, sebagai contoh data yang dipilih terlihat tersedia dari tahun 2013-hingga saat ini.

Anda harus mengecek tipe data yang sudah anda pilih, apakah sudah sesuai level datanya (raw data atau telah terkalibrasi) serta dari segi periode waktu perekaman sudah memenuhi kebutuhan anda atau belum. Bila sudah sesuai, anda dapat mencatat

imagecollection ID -nya. Sebagaimana data *Landsat 8 Collection 1 Tier 1 and Real-Time data Raw Scenes* memiliki KODE ID: **LANDSAT/LC08/C01/T1_RT**.

Setiap dataset, memiliki kode *imagecollection* ID yang unik. Anda dapat memanggil dan mengolah dataset yang diinginkan dengan cukup menginput kode *imageCollection ID* dataset tersebut didalam code editor.

USGS Landsat 8 Collection 1 Tier 1 and Real-Time data Raw Scenes

Landsat 8 Collection 1 Tier 1 and Real-Time data DN values, representing scaled, calibrated at-sensor radiance.

Landsat scenes with the highest available data quality are placed into Tier 1 and are considered suitable for time-series processing analysis. Tier 1 includes Level-1 Precision Terrain (L1TP) processed data that have well-characterized radiometry and are inter-calibrated across the different Landsat sensors. The georegistration of Tier 1 scenes will be consistent and within prescribed tolerances [≤ 12 m root mean square error (RMSE)]. All Tier 1 Landsat data can be considered consistent and inter-calibrated (regardless of sensor) across the full collection. See more information [in the USGS docs](#).

The T1_RT collection contains both Tier 1 and Real-Time (RT) assets. Newly-acquired Landsat 7 ETM+ and Landsat 8 OLI/TIRS data are processed upon downlink but use predicted ephemeris, initial bumper mode parameters, or initial TIRS line of sight model parameters. The data is placed in the Real-Time tier and made available for immediate download. Once the data have been reprocessed with definitive ephemeris, updated bumper mode parameters and refined TIRS parameters, the products are transitioned to either Tier 1 or Tier 2 and removed from the Real-Time tier. The transition delay from Real-Time to Tier 1 or Tier 2 is between 14 and 26 days.

For more information, see [Landsat 8 Handbook](#)

Data availability (time)
Apr 11, 2013 - Feb 1, 2020
Provider
USGS
Tags
landsat, usgs, global, olli-tirs, l8, lc8, c1, tier1, t1, nrt, rt, radiance
ImageCollection ID
LANDSAT/LC08/C01/T1_RT

Ketersediaan data berdasarkan tanggal perekamanan

Kode ID dataset

CLOSE

IMPORT

Scroll ke bawah anda akan menemukan kode angka dari layer BSQ dan *image properties*. Kode angka BSQ umumnya digunakan untuk membolongi (*masking*) citra misalkan berdasarkan tutupan awannya atau kadar awan tipisnya. Sedangkan *image properties* biasa digunakan untuk filter data berdasarkan tanggal perekaman, band, dll.

- Bit 1: Terrain Occlusion
 - 0: No
 - 1: Yes
- Bits 2-3: Radiometric Saturation
 - 0: No bands contain saturation
 - 1: 1-2 bands contain saturation
 - 2: 3-4 bands contain saturation
 - 3: 5 or more bands contain saturation
- Bit 4: Cloud
 - 0: No
 - 1: Yes
- Bits 5-6: Cloud Confidence
 - 0: Not Determined / Condition does not exist.
 - 1: Low, (0-33 percent confidence)
 - 2: Medium, (34-66 percent confidence)
 - 3: High, (67-100 percent confidence)
- Bits 7-8: Cloud Shadow Confidence
 - 0: Not Determined / Condition does not exist.
 - 1: Low, (0-33 percent confidence)
 - 2: Medium, (34-66 percent confidence)
 - 3: High, (67-100 percent confidence)
- Bits 9-10: Snow / Ice Confidence
 - 0: Not Determined / Condition does not exist.
 - 1: Low, (0-33 percent confidence)
 - 2: Medium, (34-66 percent confidence)
 - 3: High, (67-100 percent confidence)

Image Properties

Name	Type	Description
BPF_NAME_OLI	STRING	The file name for the Bias Parameter File (BPF) used to generate the product, if applicable. This only applies to products that contain OLI bands.
BPF_NAME_TIRS	STRING	The file name for the Bias Parameter File (BPF) used to generate the product, if applicable. This only applies to products that contain TIRS bands.
CLOUD_COVER	DOUBLE	Percentage cloud cover, -1 = not calculated.
CLOUD_COVER_LAND	DOUBLE	Percentage cloud cover over land, -1 = not calculated.
COLLECTION_CATEGORY	STRING	Tier of scene. (T1 or T2)
COLLECTION_NUMBER	DOUBLE	Number of collection.
CPF_NAME	STRING	Calibration parameter file name.
DATA_TYPE	STRING	Data type identifier. (L1T or L1G)
DATE_ACQUIRED	STRING	Image acquisition date. "YYYY-MM-DD"
DATUM	STRING	Datum used in image creation.
EARTH_SUN_DISTANCE	NUMBER	Earth sun distance in astronomical units (AU)

CLOSE

IMPORT

BAB 3 GELOMBANG ELEKTROMAGNETIK

Warna serta Rona yang terdapat pada gambar foto satelit terbentuk atas dasar interaksi obyek di permukaan bumi terhadap gelombang elektromagnetik. Sama seperti mata manusia, mekanisme terbentuknya gambar pada citra satelit tak lepas dari peran penting gelombang elektromagnetik. Matahari sebagai sumber energi bumi memancarkan gelombang elektromagnetik, dan berinteraksi dengan obyek-obyek dimuka bumi. Obyek-obyek dimuka bumi ini tersusun dari mineral-mineral yang memiliki karakteristik unit saat berinteraksi dengan gelombang elektromagnetik dari matahari. Warna hijau pada vegetasi, warna biru pada lautan luas serta warna-warna lain pada obyek yang terlihat oleh mata kita disebabkan karena obyek-obyek tersebut memiliki karakteristik material yang berbeda-beda. Warna-warna obyek tersebut pun dapat berubah bila gelombang elektromagnetik yang digunakan pada spektrum yang berbeda.

Pada bab ini, anda akan belajar men-visualisasikan citra satelit dan memahami cara terbentuknya gambaran citra satelit.

3.1. Memunculkan Single Scenes – Greyscale (studi kasus: Landsat 8 Raw)

Bila sebelumnya anda sudah berhasil melakukan *marking* lokasi dan mencari dataset, maka pada sesi ini, anda akan diajarkan membuka data satelit melalui script dasar.

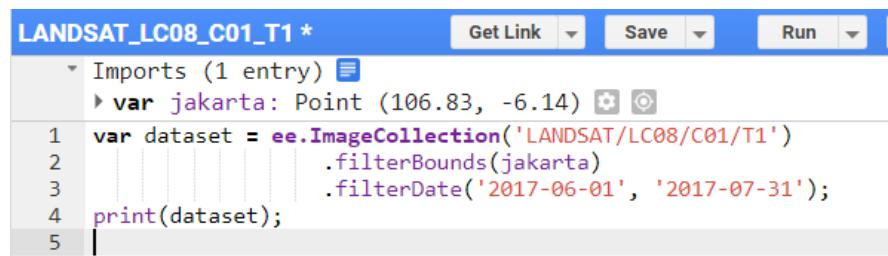
```
LANDSAT_LC08_C01_T1 *
Get Link Save Run
Imports (1 entry)
var jakarta: Point (106.83, -6.14)
var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
.filterBounds(jakarta)
.filterDate('2017-06-01', '2017-07-31');
```

Gunakan script diatas untuk memunculkan citra landsat 8 dengan jenis *Landsat 8 Collection 1 Tier 1 and Real-Time data Raw Scenes*, dengan diawali dengan 'var **dataset**', sebagaimana nama variabel datanya. Lalu tambahkan ".filterBounds" yang berfungsi untuk mencari citra di lokasi yang diinginkan. Karena sebelumnya, kita sudah plot lokasi titik (var jakarta), maka tambahkan didalam kolom .filterBounds(jakarta). Selanjutnya, tambahkan

".filterDate" dengan diikuti dengan periode tanggal perekaman yang diinginkan. Untuk mengakhiri pengaturan dari variabe **dataset**, gunakan tanda titik koma (,).

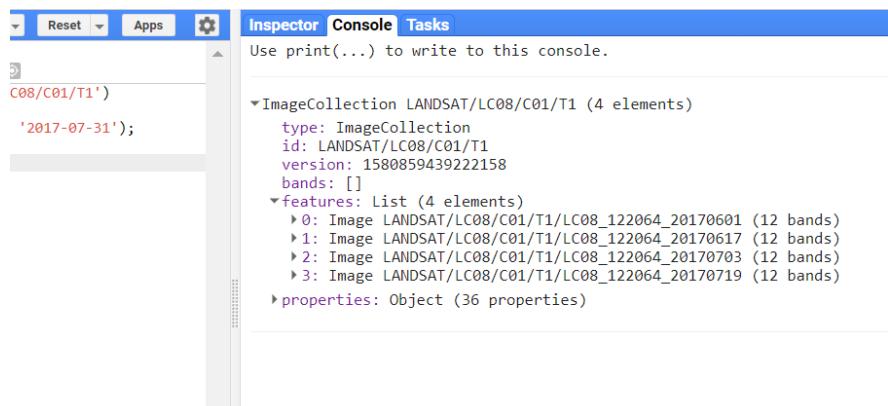
Maka dengan ini, kita sudah memilih (selected) variabel **dataset** citra Landsat 8, perekaman di lokasi jakarta dengan periode perekaman juni hingga juli 2017.

Gunakan script "print()" dengan diikuti "dataset" didalam kurungnya, untuk menunjukkan list ketersediaan scenes dari dataset yang sudah kita pilih (dataset). Lalu akhiri dengan tanda titik-koma (,).



```
LANDSAT_LC08_C01_T1 *
Get Link Save Run
Imports (1 entry)
var jakarta: Point (106.83, -6.14)
1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5 |
```

Selanjutnya, klik tombol run, maka akan muncul di tab *console* yakni berupa list scenes yang masuk sesuai kategori dataset yang kita cari.



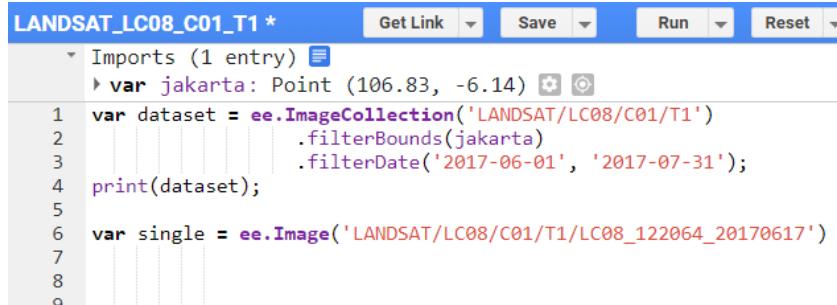
Inspector Console Tasks

Use print(...) to write to this console.

```
ImageCollection LANDSAT/LC08/C01/T1 (4 elements)
  type: ImageCollection
  id: LANDSAT/LC08/C01/T1
  version: 1580859439222158
  bands: []
  features: List (4 elements)
    > 0: Image LANDSAT/LC08/C01/T1/LC08_122064_20170601 (12 bands)
    > 1: Image LANDSAT/LC08/C01/T1/LC08_122064_20170617 (12 bands)
    > 2: Image LANDSAT/LC08/C01/T1/LC08_122064_20170703 (12 bands)
    > 3: Image LANDSAT/LC08/C01/T1/LC08_122064_20170719 (12 bands)
  properties: Object (36 properties)
```

Terlihat scenes yang masuk kategori dataset berjumlah 4 scenes. Selanjutnya anda dapat memilih dari keempat scenes tersebut dengan dengan membuat variabel baru. Sebagai contoh, kita akan membuat variabel baru var **single** dimana variabel dataset single ini merupakan data landsat 8 dengan ID scenes yang kita pilih adalah 'LANDSAT/LC08/C01/T1/LC08_122064_20170617' atau bila dilihat dari kode 20170617 yang merupakan kode tanggal perekaman, maka data ini merupakan perekaman tahun 2017, bulan Juni tanggal 17.

Anda dapat mengganti scenes sesuai tanggal perekaman yang anda inginkan asalkan scenes dengan tanggal perekaman tersebut tersedia karena Landsat 8 merekam tiap 18 hari sekali.



```

LANDSAT_LC08_C01_T1 *
Get Link Save Run Reset
Imports (1 entry)
var jakarta: Point (106.83, -6.14) ⚙️ ⚙️
1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
7
8
9

```

Selanjutnya pilih band atau saluran yang ingin dimunculkan. Sebagai informasi, Landsat 8 membawa 2 sensor OLI dan TIRS yang memiliki 11 band atau saluran. Sensor ini terdiri dari 4 band spektrum visible, 3 band spekturm IR, 1 band pankromatik, 1 band cirrus, dan 2 band thermal IR.

Landsat 8 Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS)	Bands	Wavelength (micrometers)
Launched February 11, 2013	Band 1 - Coastal aerosol	0.43 - 0.45
	Band 2 - Blue	0.45 - 0.51
	Band 3 - Green	0.53 - 0.59
	Band 4 - Red	0.64 - 0.67
	Band 5 - Near Infrared (NIR)	0.85 - 0.88
	Band 6 - SWIR 1	1.57 - 1.65
	Band 7 - SWIR 2	2.11 - 2.29
	Band 8 - Panchromatic	0.50 - 0.68
	Band 9 - Cirrus	1.36 - 1.38
	Band 10 - Thermal Infrared (TIRS) 1	10.60 - 11.19
	Band 11 - Thermal Infrared (TIRS) 2	11.50 - 12.51

Sebagai contoh, munculkan band IR atau band 5 dengan menambahkan fungsi `.select()`, dengan mengisi 'B5' didalam kurungnya pada variabel `single`. Selanjutnya gunakan tanda titik-koma (;), untuk mengakhiri pengaturan dari variabel **single**.

```

LANDSAT_LC08_C01_T1 *
Get Link Save Run Reset Apps
Imports (1 entry) ↗
var jakarta: Point (106.83, -6.14) ⚙️ ⓘ
1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
7   .select('B5');

```

Sampai pada saat ini, kita sudah punya 2 variabel, yakni var **dataset** dan **single**. Selanjutnya kita akan memunculkan (display) tampilan variabel **single** dengan hanya memunculkan band NIR (Band 5) saja.

Untuk memunculkan (display) citra, maka anda perlu mendefinisikan parameter visualisasinya. Data Landsat 8 ini memiliki resolusi radiometrik 16 bit, artinya memiliki rentang nilai piksel $0-2^{16}$, atau $0 - 65536$. Buatlah script sebagai variabel untuk menvisualisasikan citra yang berisi parameter nilai minimum dan maksimum piksel serta palet warnanya.

```

LANDSAT_LC08_C01_T1 *
Get Link Save Run Reset Apps
Imports (1 entry) ↗
var jakarta: Point (106.83, -6.14) ⚙️ ⓘ
1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
7   .select('B5');
8
9 //memunculkan citra
10 var greyscale = {min: 0, max: 65536, palette: ['black', 'white']};
11 Map.addLayer(single, greyscale, 'NIR');
12

```

Kita definisikan variabel **greyscale** dimana nilai min = 0 dan maks=65536 dimana warnanya hitam-putih (black-white). Selanjutnya, gunakan script `Map.addLayer`, dimana var **single** sebagai sumber datanya, lalu **greyscale** sebagai parameter visualnya, dan 'NIR' sebagai nama layernya. Klik *Run* untuk memunculkan image-nya.

```

Google Earth Engine Search places and datasets...
LANDSAT_LC08_C01_T1* Get Link Save Run Reset Apps Inspector Console Tasks
Click on the map to inspect the layers.

Imports (1 entry)
var jakarta: Point (106.83, -6.14) + ○
1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
7   .select('B5');
8
9 //memunculkan citra
10 var greyscale = {min: 0, max: 65535, palette: ['black', 'white']};
11 Map.addLayer(single, greyscale, 'NIR');
12

```

Bila tampilannya agak gelap, anda dapat mengedit var **greyscale** dengan menurunkan nilai maksimumnya, misalkan kita turunkan di angka 30.000.

```

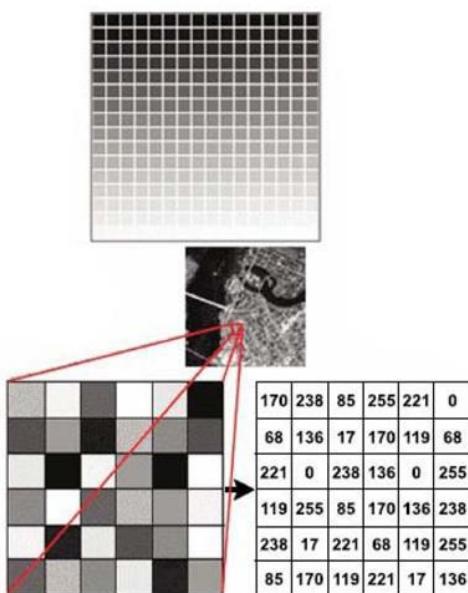
LANDSAT_LC08_C01_T1* Get Link Save Run Reset Apps Inspector Console Tasks
Imports (1 entry)
var jakarta: Point (106.83, -6.14) + ○
1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
7   .select('B5');
8
9 //memunculkan citra
10 var greyscale = {min: 0, max: 30000, palette: ['black', 'white']};
11 Map.addLayer(single, greyscale, 'NIR');
12

```

Selanjutnya, anda dapat observasi nilai piksel dengan mengaktifkan cursor ⓘ lalu carilah obyek gelap semisal badan air (laut, sungai, dan danau) lalu klik piksel obyek tersebut. Kemudian cek di kolom *inspector*, maka terlihat nilai piksel dari obyek yang anda klik.



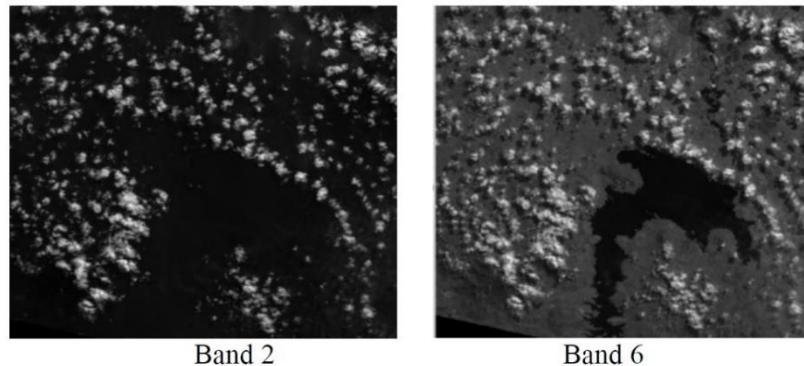
Nilai piksel (*Digital number*) pada citra satelit optis, merupakan representatif dari tingkat reflektan/radiasi suatu obyek di permukaan bumi. Seperti yang telah dijelaskan pada Bab 1, mengenai skema reflektan/radiasi proses pengambilan data dalam *remote sensing*, dimana satelit merekam tingkat reflektan/radiasi dari obyek di permukaan bumi, kemudian diterjemahkan sebagai angka-angka. Angka-angka tersebutlah yang dimaksudkan sebagai nilai piksel (*Pixel value*) atau *digital number*. *Pixel Value* ini kemudian divisualisasikan menjadi tingkat kecerahan/ derajat keabuan sehingga terlihat menjadi suatu gambar ber-rona. Rentang nilai piksel bergantung dari resolusi radiometrik sensor yang terpasang di setiap citra satelit. Sebagai contoh, Citra Landsat 7, memiliki resolusi radiometrik **8 bit**, dimana rentangnya $2^8 = 256$, sehingga nilai pikselnya adalah dari **0 hingga 255**. Sedangkan Landsat 8, memiliki resolusi radiometrik **16 bit**, artinya mampu membedakan obyek sebanyak 2^{16} , yakni 65.536 tingkat perbedaan. Hal inilah yang menyebabkan nilai piksel/digital number pada Landsat 8 memiliki nilai dalam rentang **0 hingga 65.535**.



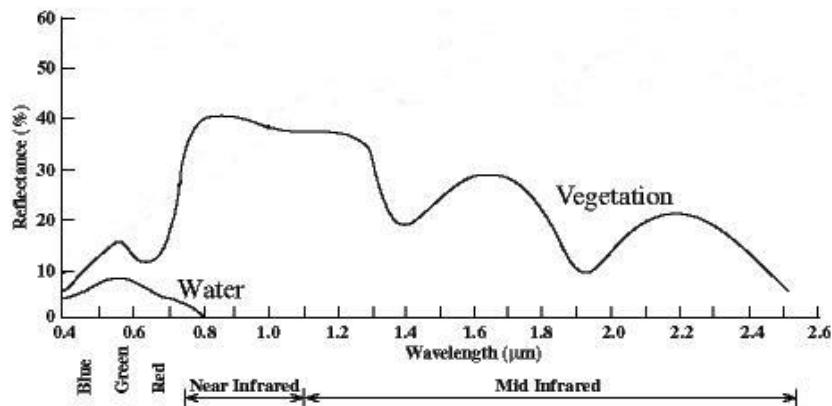
Gambar 3.1. Hubungan nilai digital number dengan derajat keabuan.

Bila kita ingin menarik garis pantai, band apa sebaiknya yang digunakan?

Sebaiknya gunakan band yang berada pada panjang gelombang inframerah dekat (NIR). Pada Landsat 8, panjang gelombang inframerah dekat masuk dalam *range* band 6 (0,85 – 0,88 μm). Perhatikanlah perbedaan kenampakan band 2 dan band 6 pada Landsat 8.



Kita dapat dengan mudah mengidentifikasi batas danau pada band 6 dibandingkan band 2 yang terlihat gelap secara keseluruhan. Hal ini disebabkan karena obyek dimuka bumi memiliki karakteristik reflektan yang berbeda. Contoh di bawah ini adalah kurva yang menunjukkan pola reflektan antara vegetasi dan air.



Pada gelombang Hijau (0,4 -0,5 μm), obyek air dan vegetasi memiliki reflektan yang sama-sama rendah, oleh karena itu, pada band 2 (0,45-0,51 μm) yang masuk kategori gelombang biru, baik air maupun daratan akan terlihat sama-sama gelap. Sedangkan pada gelombang inframerah dekat (0,7 – 1 μm), terjadi perbedaan (Gap) reflektan yang besar, di mana air memiliki reflektan sangat rendah, sedangkan vegetasi memiliki reflektan yang tinggi. Itulah mengapa pada band 6, rona vegetasi/daratan terlihat lebih terang, kontras dengan rona air yang sangat gelap sehingga dapat memudahkan kita menarik batas antara obyek air dengan obyek non-air.

Anda dapat membuat tampilan multi layer. Misalkan anda ingin menvisualisasikan band 5 dan band 3 secara bersamaan. Untuk itu, anda harus mendefinisikan variable khusus band 5 (var **singleB5**) dan band 3 (var **singleB3**), berikut scriptnya:

```

LANDSAT_LC08_C01_T1 *
Get Link Save Run Reset Apps Inspector Console Tasks
Imports (1 entry)
var jakarta: Point (106.83, -6.14) ⚙️ ⓘ
1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
7
8 var singleB5 = single.select('B5');
9 var singleB3 = single.select('B3');

```

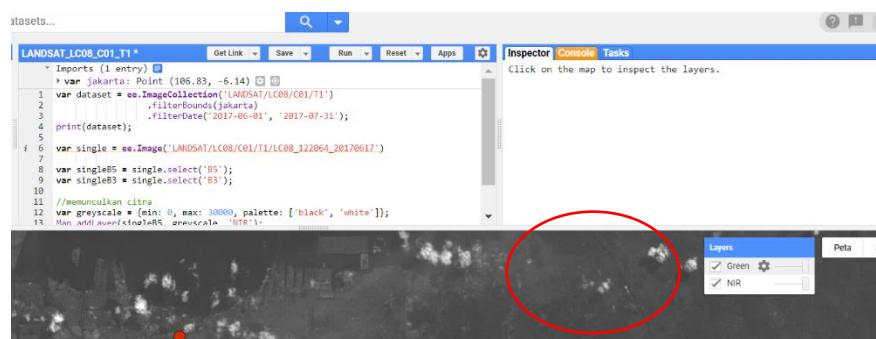
Lalu lanjutkan dengan mengedit dan menambahkan fungsi Map.addLayer untuk menampilkan (display) 2 variabel tersebut. Berikut scriptnya:

```

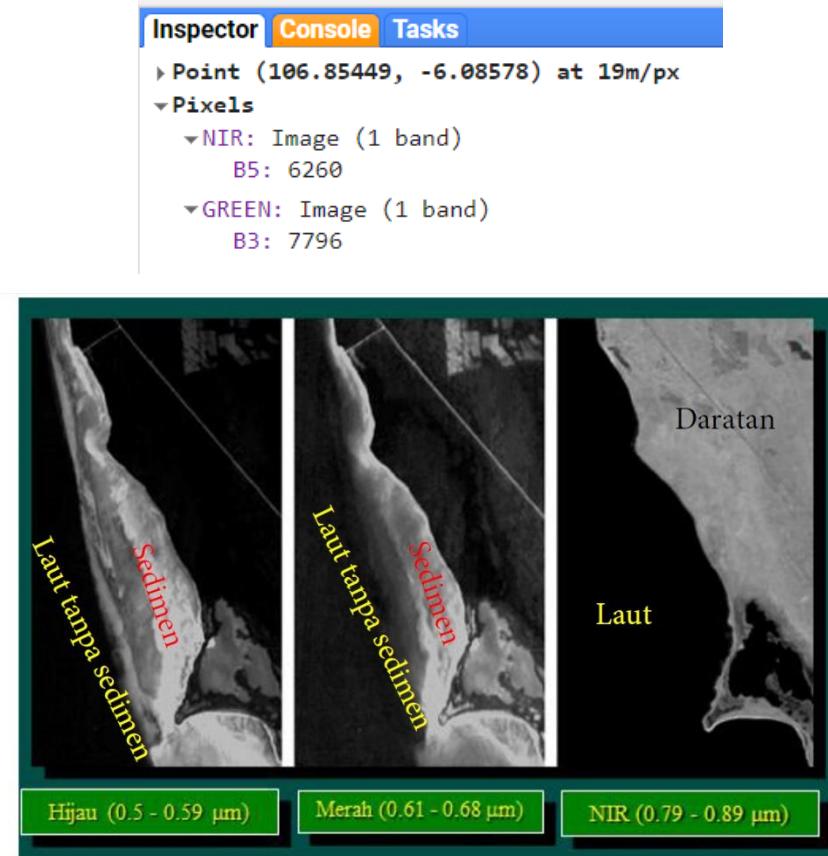
Imports (1 entry)
var jakarta: Point (106.83, -6.14) ⚙️ ⓘ
1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
7
8 var singleB5 = single.select('B5');
9 var singleB3 = single.select('B3');
10
11 //memunculkan citra
12 var greyscale = {min: 0, max: 30000, palette: ['black', 'white']};
13 Map.addLayer(singleB5, greyscale, 'NIR');
14 Map.addLayer(singleB3, greyscale, 'Green');
15

```

Kemudian klik tombol run, maka akan muncul 2 layer, yakni layer NIR dan layer Green. Anda dapat menceklis *layer box* untuk memunculkan kedua layer tersebut secara bergantian.



Sekarang lakukan hal yang sama, yakni meng-klik piksel air, lalu cek nilai piksel pada kolom *inspector*. Terlihat nilai piksel muncul pada kedua band (B5 dan B3) dimana nilai piksel berbeda walaupun obyek yang diklik sama.



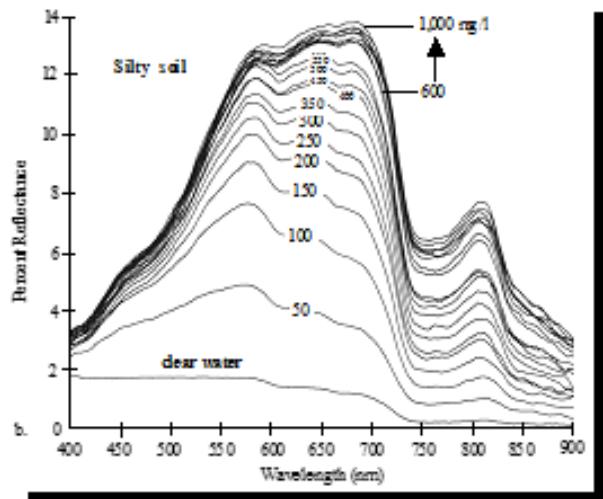
Gambar di atas menunjukkan visualisasi pada obyek air dilihat pada band (panjang gelombang) yang berbeda yakni spektrum Hijau, Merah, dan NIR. Pada spektrum NIR (gambar kanan), kita melihat laut dan darat kontras warnanya, hal ini dikarenakan spektrum NIR diserap sangat kuat oleh air sehingga warnanya gelap sedangkan darat yang berisi vegetasi memantulkan gelombang NIR sehingga warna piksel lebih cerah.

Namun saat menggunakan Band pada spektrum Merah (gambar tengah), anda akan melihat laut tersegmentasi 2 warna. Bahkan bila menggunakan spektrum hijau (gambar kiri), segmentasi warna laut sangat jelas terlihat.

Laut yang bewarna cerah merupakan air dengan kandungan sedimen dimana memiliki reflektan lebih tinggi dibandingkan laut yang tidak memiliki sedimen. Akibatnya rona air yang mengandung sedimen lebih cerah dibandingkan air yang tidak mengandung sedimen yang

berona gelap. Hal ini dapat terlihat pada spektrum merah dan lebih jelas di spektrum hijau. Namun pada spektrum NIR, air yang mengandung sedimen dan air yang tidak mengandung sedimen tetap diserap kuat sehingga warnanya sama-sama gelap.

Kurva di bawah ini menunjukkan pergeseran puncak reflektan berdasarkan kandungan sedimen dalam air. Semakin tinggi kandungan sedimen, semakin tinggi (meningkat) pula reflektan pada gelombang hijau (500 – 600 nm) dan merah (600 – 700 nm). Sedangkan pada gelombang NIR (>700 nm) dan spektrum biru (400-500 nm), peningkatan kurva reflektan tidak terlalu signifikan. Itulah sebabnya untuk mendeteksi sedimen, band yang digunakan adalah band yang berada pada spektrum merah dan hijau.



3.2. Memunculkan Single Scenes – RGB (studi kasus: Landsat 8 Raw)

Pada langkah ini, kita akan membuat tampilan citra berwarna. Untuk menampilkan citra berwarna atau citra komposit, maka dibutuhkan minimal 3 layer sekaligus yang digunakan untuk mengisi kanal Red, Green, dan Blue.

Dalam menampilkan citra satelit, terdapat 2 jenis komposit, yakni:

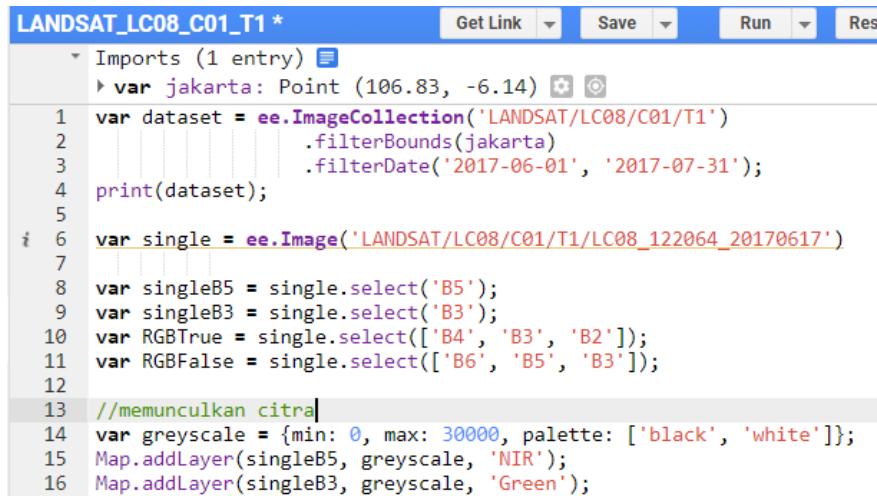
- True Color Composite (Warna sebenarnya)

True Color ini menampilkan kenampakan citra satelit yang sesuai dengan warna sebenarnya. Dalam membuat komposit warna sebenarnya ini, kita harus memasukkan band-band sesuai dengan panjang gelombangnya, misalkan kita tempatkan band merah pada kanal Red, band hijau pada kanal Green, dan band biru pada kanal Blue.

- False Color Composite (Warna semu)

False Color merupakan kombinasi RGB yang memberikan kenampakan warna obyek yang bukan sebenarnya. Biasanya komposit ini digunakan untuk penajaman visual, dengan menggunakan komposit yang tepat, maka obyek dapat terlihat lebih jelas dan kontras.

Kita akan menambahkan variabel baru pada script sebelumnya, yakni dengan var **RGBTrue** dan **RGBFalse**.



```
LANDSAT_LC08_C01_T1 *
Get Link Save Run Res
Imports (1 entry)
var jakarta: Point (106.83, -6.14)
1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
7
8 var singleB5 = single.select('B5');
9 var singleB3 = single.select('B3');
10 var RGBTrue = single.select(['B4', 'B3', 'B2']);
11 var RGBFalse = single.select(['B6', 'B5', 'B3']);
12
13 //memunculkan citra
14 var greyscale = {min: 0, max: 30000, palette: ['black', 'white']};
15 Map.addLayer(singleB5, greyscale, 'NIR');
16 Map.addLayer(singleB3, greyscale, 'Green');
```

Selanjutnya, kita menambahkan var paramater visualisasi khusus RGB. Var **greyscale** tidak bisa digunakan untuk tampilan RGB, melainkan khusus hanya untuk tampilan single image – greyscale. Untuk tampilan RGB, cukup untuk menentukan nilai min-max, tanpa perlu mendefinisikan warna (palet).

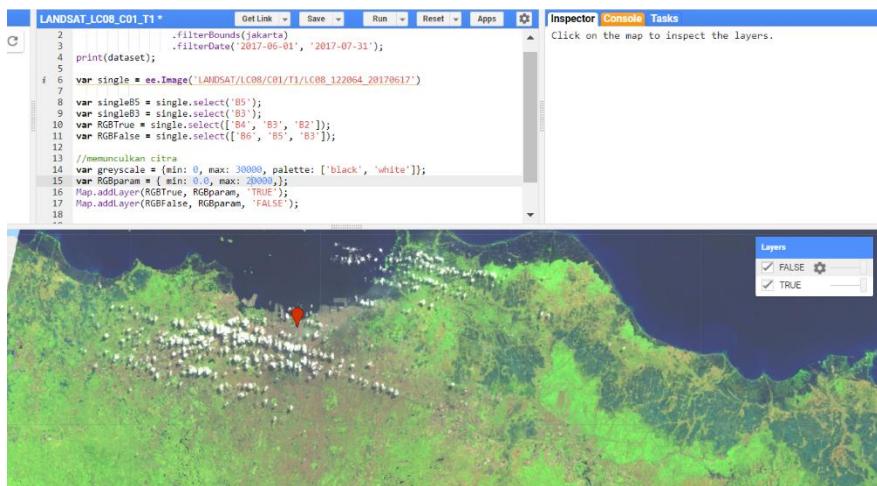
Berikut adalah script untuk menvisualisasikan tampilan RGB, yakni var **RGBparam**. Selanjutnya editlah script pada *Map.addLayer* dengan menambahkan unsur var **RGBTrue** dan **RGBFalse** sebagai input datasetnya, lalu var **RGBparam** sebagai parameter visualisasinya.

```

Imports (1 entry) ▾
▶ var jakarta: Point (106.83, -6.14) ⚙️ ⓘ
1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
7
8 var singleB5 = single.select('B5');
9 var singleB3 = single.select('B3');
10 var RGBTrue = single.select(['B4', 'B3', 'B2']);
11 var RGBFalse = single.select(['B6', 'B5', 'B3']);
12
13 //memunculkan citra
14 var greyscale = {min: 0, max: 30000, palette: ['black', 'white']};
15 var RGBparam = { min: 0.0, max: 20000,};
16 Map.addLayer(RGBTrue, RGBparam, 'TRUE');
17 Map.addLayer(RGBFalse, RGBparam, 'FALSE');
18
19

```

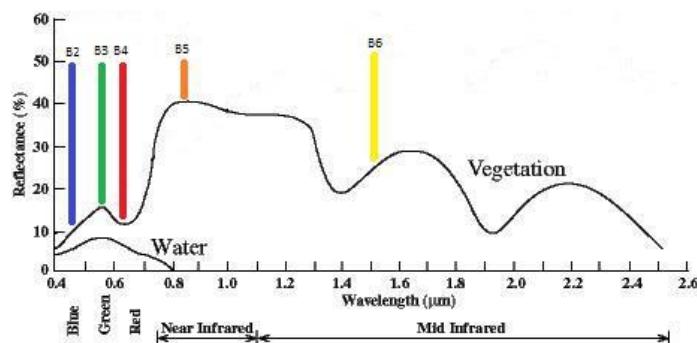
Sehingga setelah di run, maka akan muncul tampilan RGB true dan false color komposit.



Mata manusia sama halnya dengan sensor dalam satelit. Bedanya, mata manusia didesain hanya untuk menangkap gelombang tampak (*visible*), yakni dalam rentang 0,4 - 0,7 μm . Itulah sebabnya, kenampakan yang dilihat oleh mata kita sama seperti kenampakan *True Color Composite* pada citra satelit.

False color composite biasanya digunakan untuk mempertajam visualisasi citra. Misalkan dalam membedakan obyek air (sungai, danau, dan laut) dengan daratan. *False Color* RGB 653 sangat baik dalam membedakan antara obyek air dengan obyek daratan dibandingkan *True Color* RGB 432. Hal ini disebabkan karena penggunaan band 6, band 5 dan band 3 pada *False Color* merupakan band yang memiliki perbedaan (GAP) reflektan yang besar antara obyek

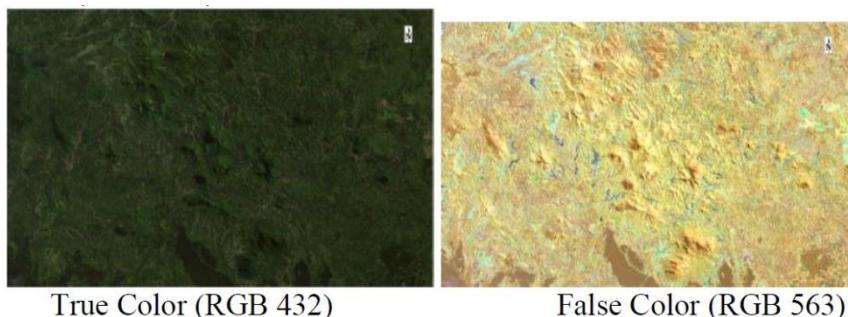
vegetasi (darat) dan laut. Berikut ini adalah grafik reflektan vegetasi dan air berdasarkan panjang gelombang serta band-band yang terdapat di Landsat 8.



Oleh karena itu, warna air dengan darat akan terlihat kontras seperti gambar di bawah ini.



Kemudian visualisasi efek topografi sangat baik ditampilkan pada komposit RGB 563 (Landsat 8).



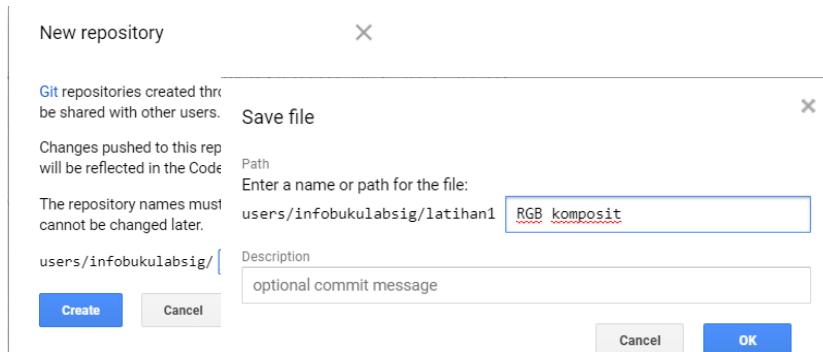
Anda dapat menyimpan script dengan cara klik save as

```

1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-06-17')
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
7
8 var singleB5 = single.select('B5');
9 var singleB3 = single.select('B3');
10 var RGBTrue = single.select(['B4', 'B3', 'B2']);
11 var RGBFalse = single.select(['B6', 'B5', 'B3']);
12
13 //memunculkan citra

```

Kemudian buatlah repositori atau folder baru. Kemudian simpan scriptnya didalam folder repositori yang dinginkan.



Script anda tersimpan didalam kolom scripts sesuai folder repositori yang anda buat.

3.3. Grafik Z Value (studi kasus: Landsat 8 Raw)

Pada sesi ini, kita akan memunculkan sebuah grafik reflektan untuk beberapa obyek. Langkah awal kita akan mengambil training area untuk 3 obyek, yakni air, vegetasi, dan lahan

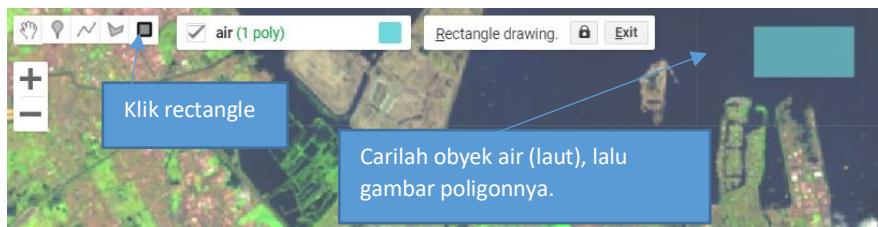
terbangun. Ketiga obyek tersebut kemudian akan dilihat nilai piksel dari band 1 hingga band 7 Citra Landsat 8 dan divisualisasikan dalam bentuk chart/grafik.

Pada sub-bab sebelumnya, anda sudah berhasil membuat tampilan citra RGB. Sekarang anda akan mencoba menggambar polygon untuk mengambil sampel piksel untuk 3 obyek, yakni air, vegetasi dan lahan terbangun.

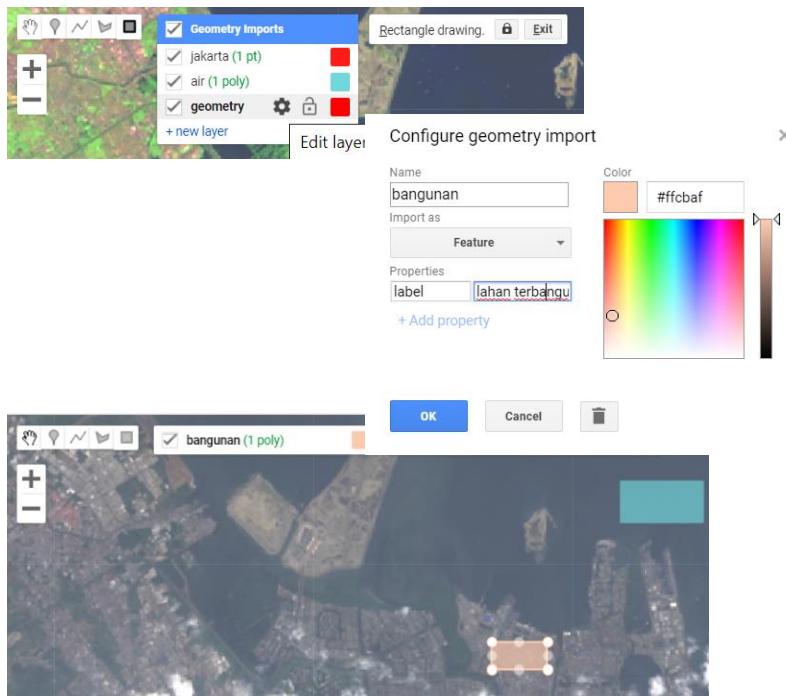
Berikut cara membuat sampel training area.



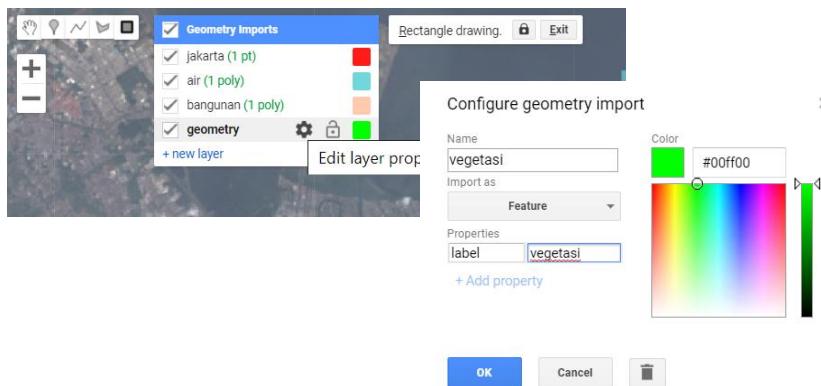
Setelah melakukan pengaturan *property* dari sampel air, selanjutnya carilah obyek air pada citra, usahakan munculkan dahulu citra RGB-nya, untuk mencari obyek air yang diinginkan. Lalu gambarlah poligonnya, perhatikan pada saat menggambar polygon, pastikan obyek non-air tidak masuk didalam polygon training yang anda buat sehingga polygon tersebut hanya berisi piksel-piksel badan air saja.

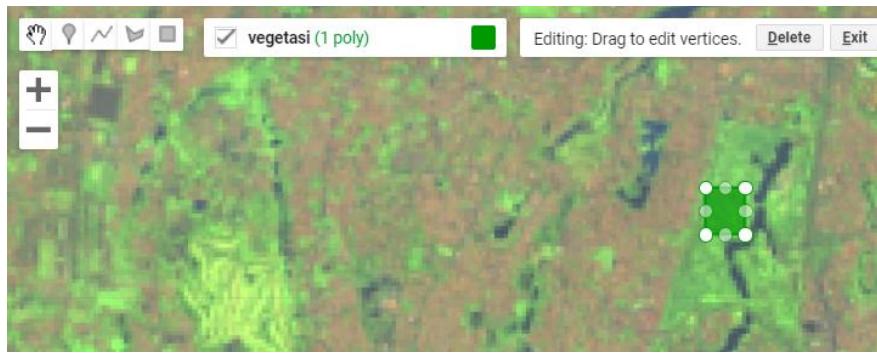


Setelah anda membuat sampel polygon air, lanjutkan dengan membuat sampel lahan terbangun.



Lanjutkan kembali dengan mengambil sampel vegetasi. Gunakan layer False color komposit untuk mempermudah pencarian obyek vegetasi yang bewarna hijau.





Bila sudah selesai dalam pengambilan training, seluruh list training area muncul pada sisi atas code editor.

New Script *

Imports (4 entries)

- ↳ var jakarta: Point (106.84, -6.13)
- ↳ var air: Feature 0 (Polygon, 1 property)
- ↳ var bangunan: Feature 0 (Polygone, 1 property)
- ↳ var vegetasi: Feature 0 (Polygon, 1 property)

```

1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
7
8 var singleB5 = single.select('B5');
9 var singleB3 = single.select('B3');
10 var RGBTrue = single.select(['B4', 'B3', 'B2']);
11 var RGBFalse = single.select(['B6', 'B5', 'B3']);
12

```

Pada script sebelumnya, anda sudah membuat var RGBTrue dan RGBFalse, dimana kedua variable masing-masing hanya terdiri dari 3 kombinasi band. Selanjutnya buatlah variable baru, yakni var **band17**, yaitu berisi citra single namun terdiri dari 7 band, yakni Band 1 hingga band 7. Untuk membuat var berisi 7 band, setelah fungsi ".select" masukkan kode 'B[1-7]'.

Setelah itu, buatlah var baru, yakni var **sampel** yang berisi sampel yang sudah anda buat (air,bangunan,vegetasi). Berikut ini adalah penulisan script 2 var tersebut, var **band17** dan **sampel**.

New Script *

Get Link Save

```

Imports (4 entries) ▾
  ↗ var jakarta: Point (106.84, -6.13)
  ↗ var air: Feature 0 (Polygon, 1 property)
  ↗ var bangunan: Feature 0 (Polygon, 1 property)
  ↗ var vegetasi: Feature 0 (Polygon, 1 property)

1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
7
8 var singleB5 = single.select('B5');
9 var singleB3 = single.select('B3');
10 var RGBTrue = single.select(['B4', 'B3', 'B2']);
11 var RGBFalse = single.select(['B6', 'B5', 'B3']);
12
13 //memunculkan citra
14 var greyscale = {min: 0, max: 30000, palette: ['black', 'white']};
15 var RGBparam = { min: 0.0, max: 20000,};
16 Map.addLayer(RGBTrue, RGBparam, 'TRUE');
17 Map.addLayer(RGBFalse, RGBparam, 'FALSE');
18
19 var band17 =single.select(['B[1-7']]);
20 var sampel = ee.FeatureCollection([air,bangunan,vegetasi]);
21

```

Selanjutnya buatlah variable baru yakni variable parameter grafik. Berilah variable ini dengan nama, misalkan var **options**. Variable ini berisi pengaturan judul grafik pada *title*, judul x dan y axis pada *hAxis* dan *vAxis*, serta pengaturan ketebalan garis dan titik pada *lineWidth* dan *pointSize*.

Berikut inilah script lengkap var **options** sebagai parameter grafik.

```

18
19 var band17 =single.select(['B[1-7']]);
20 var sampel = ee.FeatureCollection([air,bangunan,vegetasi]);
21
22 // script parameter grafik
23 var options = {
24   title: 'Grafik Nilai Piksel per Band Obyek air, vegetasi dan bangunan',
25   hAxis: {title: 'Band'},
26   vAxis: {title: 'Nilai Piksel'},
27   lineWidth: 1,
28   pointSize: 4,
29 };
30

```

Selanjutnya tambahkan variable baru, yakni variable untuk memunculkan grafik. Berilah nama variable ini, misalkan var **chart**, kemudian lakukan pengaturan semisal, **band17** sebagai input image-nya, **sampel** sebagai kumpulan training areanya.

Lalu terdapat script ee.Reducer.mean(), ini menandakan bahwa piksel yang berada pada jenis training yang sama, nilainya akan di rata-ratakan tiap bandnya.

Kemudian pada script .setOptions(), inputlah options didalam tanda kurung sebagai tanda penggunaan parameter options yang sebelumnya dibuat.

Berikut ini adalah script lengkap untuk memunculkan grafik.

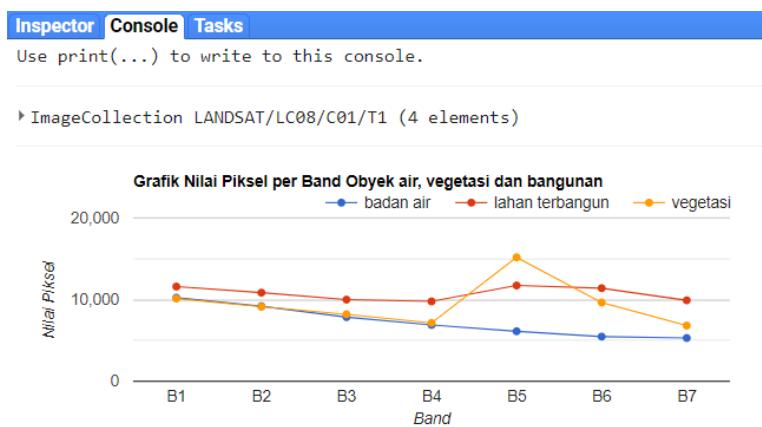
```
31 // Script memunculkan grafik
32 var chart = ui.Chart.image.regions(
33   band17, sampel, ee.Reducer.mean(), 30, 'label')
34   .setChartType('ScatterChart')
35   .setOptions(options);
36
37
```

Lalu munculkan script print(), dimana var chart sebagai inputnya. Script ini untuk memunculkan tampilan grafiknya.

```
37 // Display grafik.
38 print(chart);
39
40
```

Selanjutnya klik **run** untuk menjalankan perintah.

Maka grafik akan muncul didalam kolom *console*.



Terlihat pada grafik z profile di atas, dimana susunannya terdiri dari sumbu x (x axis) berupa band dan sumbu y (y axis) berupa nilai piksel. Umumnya grafik reflektan memiliki sumbu X berupa Panjang gelombang (wavelength) dan sumbu y berupa persentase reflektan.

Anda dapat membuat grafik z profile layaknya grafik reflektan, karena sebanarnya pada sumbu X, band dapat dikonversi dalam nilai Panjang gelombang. Anda dapat berpedoman terhadap informasi spesifikasi Panjang gelombang tiap band Landsat 8 seperti table di bawah ini.

Landsat 8 Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS)	Bands	Wavelength (micrometers)
Launched February 11, 2013	Band 1 - Coastal aerosol	0.43 - 0.45
	Band 2 - Blue	0.45 - 0.51
	Band 3 - Green	0.53 - 0.59
	Band 4 - Red	0.64 - 0.67
	Band 5 - Near Infrared (NIR)	0.85 - 0.88
	Band 6 - SWIR 1	1.57 - 1.65
	Band 7 - SWIR 2	2.11 - 2.29
	Band 8 - Panchromatic	0.50 - 0.68
	Band 9 - Cirrus	1.36 - 1.38
	Band 10 - Thermal Infrared (TIRS) 1	10.60 - 11.19
	Band 11 - Thermal Infrared (TIRS) 2	11.50 - 12.51

Sedangkan pada sumbu y, anda dapat menkonversi nilai piksel kedalam nilai persentase reflektan. Proses konversi ini sangat mudah, yakni dengan cara cukup mengganti sumber data yang awalnya berupa Landsat 8 RAW, diubah menggunakan data Landsat 8 TOA. Data Landsat 8 TOA ini adalah jenis data Landsat 8 yang sudah memiliki level lebih tinggi dimana nilai pikselnya sudah dikalibrasi kedalam nilai reflektan. Kisaran nilai pikselnya pun bukan puluhan ribu, namun bekisar antara 0 hingga 0,5.

Untuk dapat membuat grafik reflektan dimana sumbu X berupa Panjang gelombang dan sumbu Y berupa nilai reflektan, maka anda perlu melakukan perubahan pada script.

Pertama, gantilah penggunaan data Landsat 8 RAW menjadi Landsat 8 TOA, dengan mengganti script pada var **single**. Sisipkan kode **_TOA** setelah ee.Image('LANDSAT/LC08/C01/T1'.

Sehingga script var **single** menjadi seperti dibawah ini.

```

chart*
Get Link Save Run Reset Apps
* Imports (4 entries)
  var jakarta: Point (106.84, -6.13)
  var air: Feature 0 (Polygon, 1 property)
  var bangunan: Feature 0 (Polygon, 1 property)
  var vegetasi: Feature 0 (Polygon, 1 property)
1 var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1')
2   .filterBounds(jakarta)
3   .filterDate('2017-06-01', '2017-07-31');
4 print(dataset);
5
6 var single = ee.Image('LANDSAT/LC08/C01/T1_TOA/LC08_122064_20170617')
7
8 var singleB5 = single.select('B5');
9 var singleB3 = single.select('B3');
10 var RGBTrue = single.select(['B4', 'B3', 'B2']);
11 var RGBFalse = single.select(['B6', 'B5', 'B3']);
12

```

Kemudian jangan lupa mengganti var **RGBparam** atau parameter visualisasi RGB karena data yang digunakan adalah landsat 8 TOA, maka rentang nilai max hanya sampai 0.3.

```

12 //memunculkan citra
13 var greyscale = {min: 0, max: 0.3, palette: ['black', 'white']};
14 var RGBparam = { min: 0.0, max: 0.3,};
15 Map.addLayer(RGBTrue, RGBparam, 'TRUE');
16 Map.addLayer(RGBFalse, RGBparam, 'FALSE');
18

```

Selanjutnya edit script var **options** atau parameter grafik. Gantilah untuk judul, keterangan sumbu x dan sumbu y.

```

22 // script parameter grafik
23 var options = {
24   title: 'Grafik Reflektan per Band Obyek air, vegetasi dan bangunan',
25   hAxis: {title: 'Panjang Gelombang'},
26   vAxis: {title: 'Persentase Reflektan'},
27   lineWidth: 1,
28   pointSize: 4,
29 };
30

```

Selanjutnya sisipkan variabel baru, berilah nama var **wavelength** yang berisi label atau keterangan Panjang gelombang dari band 1 hingga band 7.

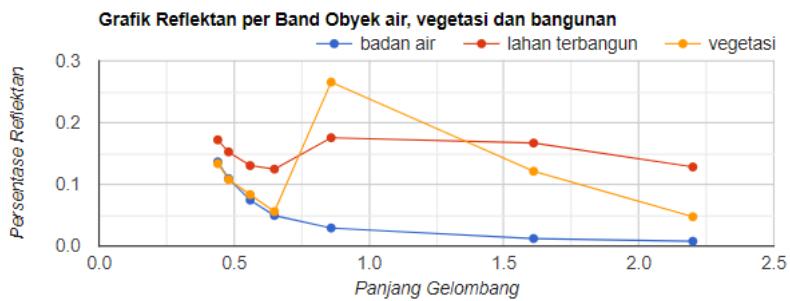
Kemudian pada var chart, masukkan var **wavelength** setelah fungsi 'label'.

```

30 var wavelengths = [0.44, 0.48, 0.56, 0.65, 0.86, 1.61, 2.2];
31
32
33 // Script memunculkan grafik
34 var chart = ui.Chart.image.regions(
35   band17, sampel, ee.Reducer.mean(), 30, 'label', wavelengths)
36   .setChartType('ScatterChart')
37   .setOptions(options);
38
39 // Display grafik.
40 print(chart);

```

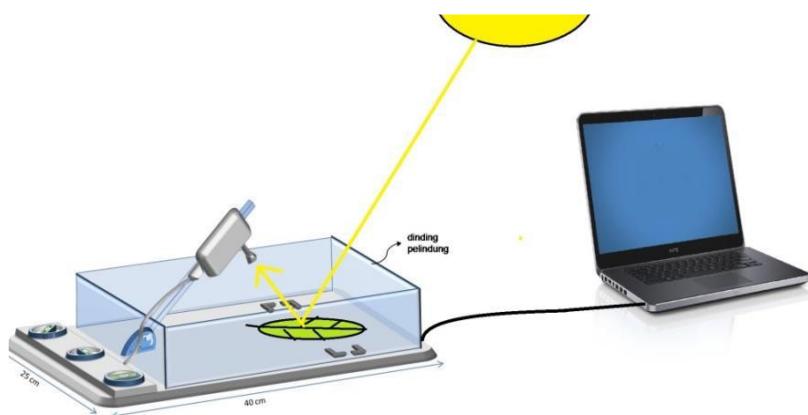
Selanjutnya klik tombol **run**



Maka grafik z profile yang muncul sudah sesuai dengan grafik reflektan pada umumnya.

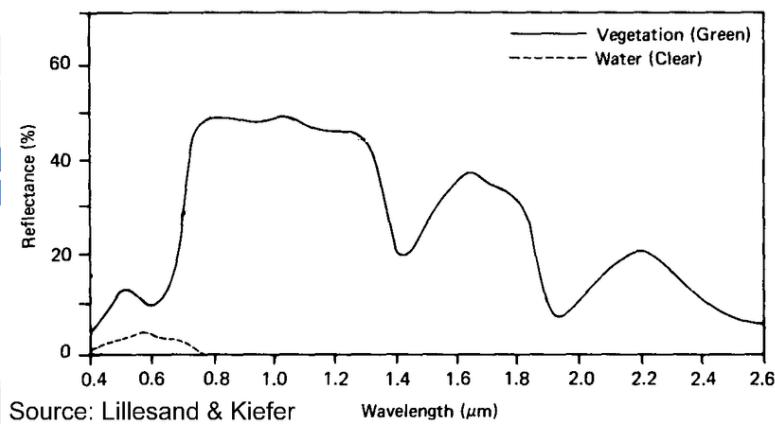
Pola reflektan object sebenarnya dapat diukur menggunakan suatu alat yang bernama **spektrometer/spectroradiometers**. Alat ini memiliki sensor layaknya sensor satelit, namun sensor yang dimiliki alat spectrometer ini umumnya berupa hyperspectral, dimana saluran panjang gelombangnya banyak dan kontinu tidak seperti sensor satelit yang pembagian saluran Panjang gelombangnya terbagi kedalam beberapa band. Alat ini tidak bisa menghasilkan gambar, namun hanya menghasilkan grafik pola reflektan.

Waktu pengukuran dalam menggunakan alat ini paling optimal adalah sekitar jam 11 pagi hingga jam 2 siang, karena pada selang waktu tersebut, intensitas gelombang elektromagnetik matahari sangat tinggi, dengan syarat cuaca tidak mendung. Obyek diletakkan tepat di depan sensor, lalu sensor diarahkan tepat mengenai obyek agar pantulannya dapat terekam. Kemudian, alat tersebut tersambung dengan PC/laptop yang kemudian digunakan untuk menyimpan *spectra library* hasil pengukurannya.



Gambar 3.1. Spektrometer

Berikut adalah contoh hasil rekaman reflektan vegetasi menggunakan alat spektrometer (atas), kemudian bandingkan dengan hasil pola reflektan vegetasi yang diambil pada citra Landsat 8 (bawah) dengan menggunakan z profil.



BAB 4 PREPROCESSING

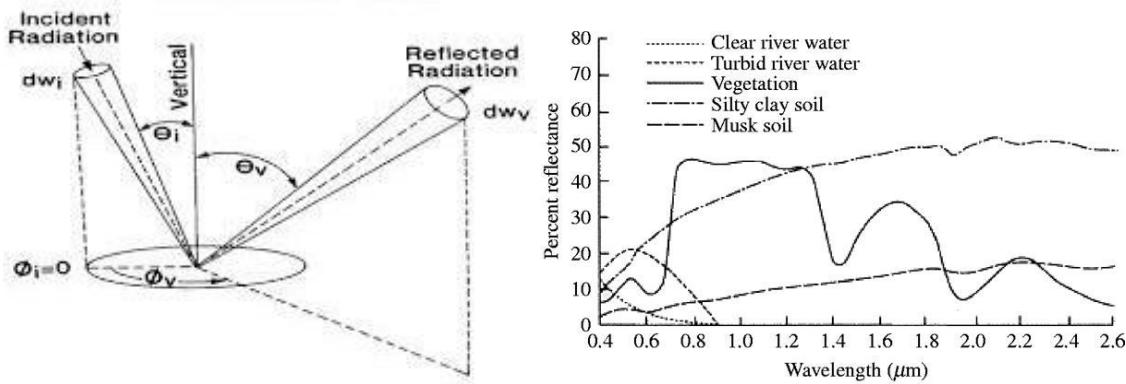
4.1. Kalibrasi Radiometrik

Proses kalibrasi merupakan proses untuk melakukan transformasi nilai piksel untuk mendapatkan nilai spektral radian dan reflektan. Citra hasil perekaman satelit yang masih original atau bersifat data mentah (Raw Data) memiliki nilai piksel berupa digital number (DN). Digital number (DN) dalam piksel dapat dikatakan indeks angka yang merepresentasikan tingkat pantulan (reflektan) gelombang elektromagnetik terhadap obyek dimuka bumi.

Proses kalibrasi ini bersifat optional, artinya tidak selalu harus dilakukan. Proses kalibrasi ini sangat penting apabila user menginginkan transformasi nilai piksel dalam bentuk nilai spektral radians dan nilai reflektan sebenarnya. Aplikasi dari penerapan kalibrasi ini misalkan kebutuhan suatu algoritma dimana input citra harus memiliki nilai piksel dalam satuan nilai spektral reflektan, contohnya adalah algoritma yang dikembangkan oleh Budiman (2005) untuk menghitung kandungan sedimen dalam air. Selain itu, untuk mendapatkan informasi nilai suhu permukaan darat, maka band thermal yang digunakan harus dikalibrasi kedalam nilai spektra radians.

Perlu dipahami bahwa terdapat perbedaan makna antara spektral radian dengan reflektan. Spektral radian adalah jumlah energi/flux yang diradiasikan oleh obyek dimuka bumi per unit luasan pada sudut tertentu. Nilai dari spektral radian ini memiliki satuan Watts/m²*srad*μm.

Sedangkan reflektan adalah persentase energi yang dipantulkan obyek dari total energi yang diterima obyek per satuan luas. Nilai reflektan ini tidak memiliki satuan dan biasanya diukur berdasarkan persentase (%). Gambar dibawah ini adalah contoh pola reflektan beberapa macam obyek.



Pada sub-bab 3.3 sebelumnya, Data Landsat 8 dengan kode ".../T1_TOA/..." adalah jenis dataset dari produk Landsat 8 yang sudah mengalami proses kalibrasi radiometrik ke nilai reflektan. Nilai piksel pada citra Landsat 8 TOA memiliki rentang sekitar 0 – 1, berbeda dengan data Landsat 8 RAW yang memiliki rentang nilai 0 - 65.536. Sehingga, bila anda menggunakan dataset Landsat 8 dengan kode (T1_TOA) maka secara tidak langsung anda sudah menggunakan data yang telah terkalibrasi.

Sedangkan pengolahan data radiometrik secara manual, maka anda harus mentransformasi seluruh piksel dengan rumus kalibrasi yang telah ada. Dalam pengolahan data Landsat 8, kalibrasi radiometrik baik reflektan maupun radience ini dapat dihitung dengan rumus sebagai berikut:

$$\text{Reflektan } \rho\lambda' = M_p * Q_{cal} + A_p$$

Dimana : Mp = faktor skala sedangkan Ap = faktor penambah

Dalam menghitung reflektan, tidak semua band dapat digunakan. Band yang memiliki panjang gelombang thermal (band 10 dan 11) tidak dapat dikonversi ke satuan reflektan, karena band ini hanya merekam radiasi saja.

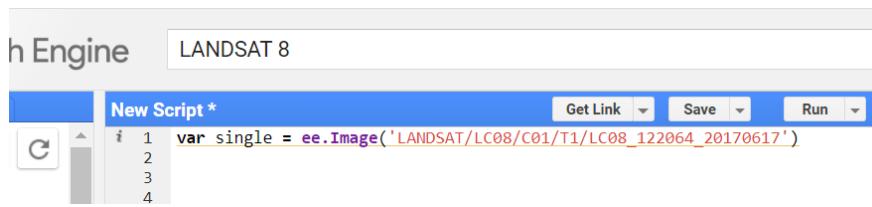
$$\text{Spektral Radian (Radiance) } L\lambda = M_L * Q_{cal} + A_L$$

Dimana : M_L = faktor skala sedangkan A_L = faktor penambah

Band reflektan (OLI) maupun thermal (TIRS) dapat dikonversi kedalam satuan nilai reflektan radian ini.

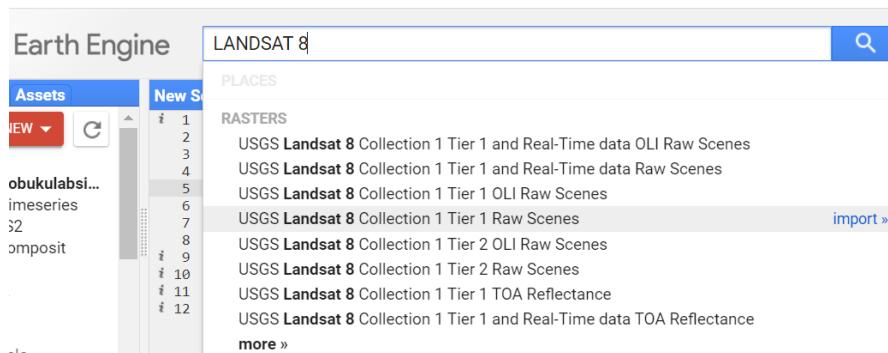
Berikut ini adalah langkah-langkah serta script untuk melakukan proses kalibrasi radiometrik.

Carilah dataset dengan tipe single image atau anda bisa gunakan data yang dipakai pada sub bab 3.3 yakni layer Landsat 8 T1 (Raw) dengan id 'LANDSAT/LC08/C01/T1/LC08_122064_20170617'. Masukkan id scenes tersebut dalam variabel baru, misalkan **var single**.

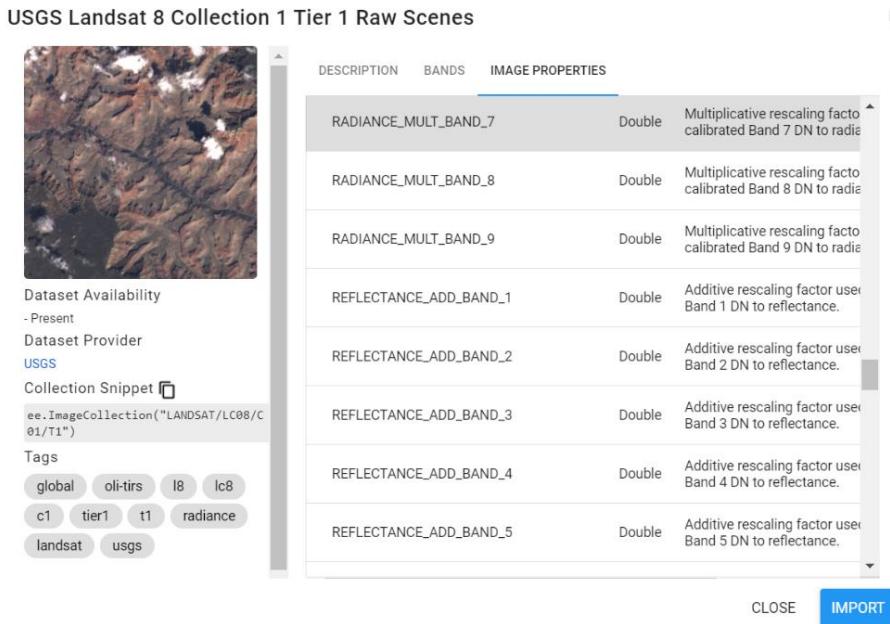


```
New Script * Get Link Save Run
1 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
2
3
4
```

Selanjutnya anda harus mencari nilai faktor skala multiplikatif (M) dan aditif (A). Untuk mencarinya, anda dapat mengecek item properties dari dataset yang digunakan. Ketiklah landsat 8 pada kolom search, lalu pilih "*USGS Landsat 8 Collection 1 Tier 1 Raw Scenes*".



Kemudian masuk kedalam tab IMAGE PROPERTIES, kemudian carilah grup parameter Radiance dan Reflektan. Anda akanapatkan kode Reflectance_Add, Reflectance_Mult, Radiance_Add, dan Radiance_Mult untuk seluruh band. Misalkan anda akan melakukan kalibrasi reflektan terhadap band 1. Maka, carilah kode parameter *REFLECTANCE_MULT_BAND_1* dan *REFLECTANCE_ADD_BAND_1*.



Buatlah 2 variabel baru, yakni misakan MIB1 dan AIB1 lalu gunakan fungsi gets diikuti kode parameter di atas. Lalu gunakan **print ()** untuk memunculkan kedua nilai parameter Band 1 tersebut.

```

New Script * Get Link Save Run Reset Apps
1 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
2
3 var AaB1 = single.get('REFLECTANCE_ADD_BAND_1');
4 var MIB1 = single.get('REFLECTANCE_MULT_BAND_1');
5
6 print (MIB1);
7 print (AaB1);
8
9
Inspector Console Tasks
Use print(...) to write to
0.00001999999494757503
-0.10000000149011612

```

Klik **Run**, lalu terlihat pada kolom *console*, nilai parameter Reflektan *Mult* Band 1 = -0,1 dan *Add* Band 1 = 0,00001999. Anda dapat mengecek nilai Reflektan *Mult* dan *Add* seluruh band. Umumnya, nilai reflektan *Mult* dan *Add* seluruh band memiliki nilai yang sama. Sehingga anda dapat melakukan proses kalibrasi kedalam nilai reflektan secara sekaligus (seluruh band).

Selanjutnya buat variabel baru yakni **var ref** yang berisi rumus kalibrasi untuk menghasilkan layer reflektan dengan mengalikan **var single** dengan nilai faktor *Mult* ditambahkan dengan nilai faktor *Add*.

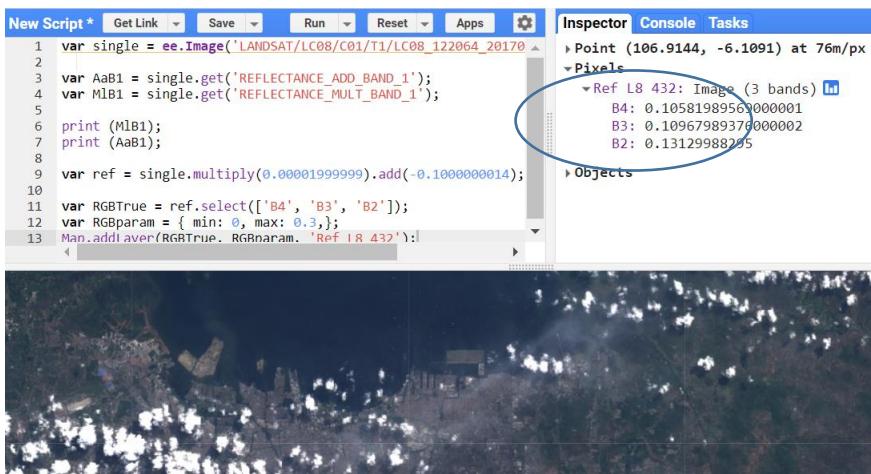
Kemudian buatkan **var RGBTrue** yang berisi seleksi 3 band terhadap layer **var ref** untuk menampilkan komposit warna sebenarnya, yakni Band 4, Band 3 dan Band 2. Kemudian tambahkan **var RGBparam** sebagai parameter visualisasi dengan pengaturan nilai minimum = 0 dan maksimum 0,3. Terakhir tambahkan fungsi *Map.addLayer*, inputlah **var RGBTrue** dan gunakan **var RGBparam** sebagai parameter, lalu berinama layer sebagai 'Ref L8 432'.

```

New Script * Get Link Save Run Reset Apps
1 var single = ee.Image('LANDSAT/LC08/C01/T1/LC08_122064_20170617')
2
3 var AaB1 = single.get('REFLECTANCE_ADD_BAND_1');
4 var MlB1 = single.get('REFLECTANCE_MULT_BAND_1');
5
6 print (MlB1);
7 print (AaB1);
8
9 var ref = single.multiply(0.0000199999).add(-0.100000014);
10
11 var RGBTrue = ref.select(['B4', 'B3', 'B2']);
12 var RGBparam = { min: 0, max: 0.3,};
13 Map.addLayer(RGBTrue, RGBparam, 'Ref L8 432');

```

Klik tombol **Run** hingga muncul citranya dimana citra tersebut adalah Landsat 8 yang sudah terkalibrasi kedalam nilai reflektan. Ceklah nilai piksel dengan cara membuka tab *inspektor*, lalu klik salah satu lokasi piksel dalam citra hingga muncul nilai reflektannya. Anda dapat menemukan nilai reflektan yang nilainya berada dikisaran 0 – 0,7



Sedangkan untuk kalibrasi kedalam nilai radiance memiliki langkah yang sama, dimana nilai factor skala multiplikatif dan additive radiance setiap band harus dicari terlebih dahulu. Cobalah untuk mengecek nilai factor skala radiance multiplikatif dan aditif untuk setiap band. Umumnya nilai mult dan add antar band memiliki nilai yang berbeda, hal ini berbeda dengan grup faktor skala reflektan.

Lanjutkan script reflektan sebelumnya, namun buatlah garis pembatas untuk membatasi antara script reflaktan dan radiance. Gunakan garis miring sebanyak 2 kali, hal ini berguna untuk menonaktifkan script atau menggolongkan text sebagai non-script. Lalu gunakan tombol minus untuk membuat garis pembatas putus-putus, lalu anda dapat tambahkan keterangan, misalkan "radiance".

```

New Script * Get Link Save Run Reset Apps
7 print('M1B1');
8
9 var ref = single.multiply(0.00001999999).add(-0.100000014)
10
11 var RGBTrue = ref.select(['B4', 'B3', 'B2']);
12 var RGBparam = { min: 0, max: 0.3,};
13 Map.addLayer(RGBTrue, RGBparam, 'Ref L8 432');
14
15 //-----radiance-----
16

```

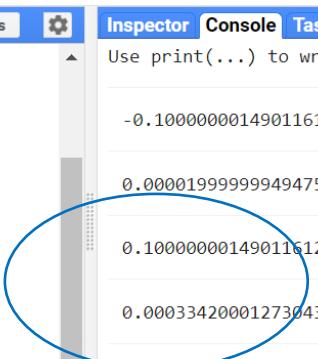
Selanjutnya buatlah 2 variabel, yakni var AdB10 dan var MdB10 untuk mendefinisikan nilai faktor radience Mult dan Add di Band10. Lalu gunakan fungsi print() untuk memunculkan kedua nilai tersebut

```

New Script * Get Link Save Run Reset
12 var RGBparam = { min: 0, max: 0.3,};
13 Map.addLayer(RGBTrue, RGBparam, 'Ref L8 432');
14
15 //-----radiance-----
16
17 var AdB10 = single.get('RADIANC_ADD_BAND_10');
18 var MdB10 = single.get('RADIANC_MULT_BAND_10');
19
20 print(AdB10);
21 print(MdB10);
22
23

```

Klik **run** untuk memunculkan nilainya.



```

Get Link Save Run Reset Apps
multiply(0.00001999999).add(-0.100000014)
select(['B4', 'B3', 'B2']);
0, max: 0.3,};
RGBparam, 'Ref L8 432');

ce-----

t('RADIANC_ADD_BAND_10');
t('RADIANC_MULT_BAND_10');

elect('B10').multiply(0.0003342).add(0.1000000149)

Inspector Console Tasks
Use print(...) to write to the console
-0.1000000149011612
0.00001999999494757503
0.1000000149011612
0.00033420001273043454

```

Terlihat pada kolom *console*, nilai faktor M B10 sebesar 0,0003342 dan nilai faktor A B10 sebesar 0,1. Selanjutnya buat var **radB10**, dimana inputkan var single diserta fungsi select (B10), agar band yang diproses hanyalah band 10. Masih di var radB10, kalikan dengan 0,0003342 sebagai faktor M dan ditambahkan 0,1 sebagai faktor A.

Selanjutnya tambahkan var **greyscale** sebagai parameter visualisasi dimana nilai min = 0 dan maks = 11, serta gunakan pallete warna *greyscale* (black-white).

Terakhir tambahkan fungsi **Map.addLayer** untuk memunculkan layernya, dimana **radB10** sebagai inputnya, **greyscale** sebagai parameter visualisasi, dan berinama layer misalkan 'RAD B10'.

Berikut adaah script lengkapnya.

```
8
9 var ref = single.multiply(0.00001999999).add(-0.1000000014)
10
11 var RGBTrue = ref.select(['B4', 'B3', 'B2']);
12 var RGBparam = { min: 0, max: 0.3,};
13 Map.addLayer(RGBTure, RGBparam, 'Ref L8 432');
14
15 //-----radiance-----
16
17 var AdB10 = single.get('RADIANCE_ADD_BAND_10');
18 var MdB10 = single.get('RADIANCE_MULT_BAND_10');
19
20 print(AdB10);
21 print(MdB10);
22
23 var radB10 = single.select('B10').multiply(0.0003342).add(0.10000000149);
24 var greyscale = {min: 8, max: 11, palette: ['black', 'white']};
25 Map.addLayer(radB10, greyscale, 'RAD B10');
```

Kemudian klik tombol **Run** layer, maka dimap view anda dapat melihat 2 layer, yakni layer 'Ref L8 432' dan layer 'RAD B10'. Anda dapat mengecek nilai piksel dari layer RAD B10, dimana kini nilainya merupakan nilai radiace dengan kisaran 0-20.

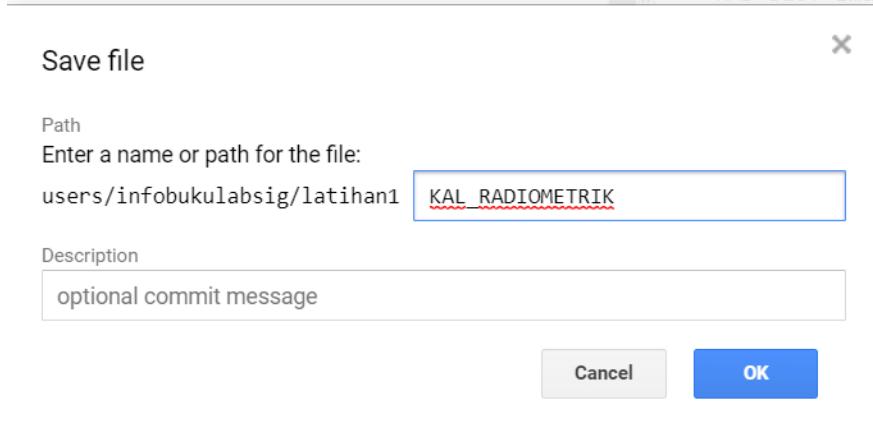
Semakin tinggi angkanya atau semakin warna putih, maka tingkat radiasi energi yang dipancarkan semakin tinggi. Umumnya, setelah mendapatkan nilai radiance B10, maka anda bisa melanjutkan dengan mentransformasi layer tersebut menjadi suhu. Topik

```

New Script * Get Link Save Run Reset Apps Inspector Console Tasks
8 var ref = single.multiply(0.0001999999).add(-0.1000000014)
9
10 var RGBTrue = ref.select(['B4', 'B3', 'B2']);
11 var RGBparam = { min: 0, max: 0.3 };
12 Map.addLayer(RGBTrue, RGBparam, 'Ref L8 432');
13
14 //-----radiance-----
15
16 var AdB10 = single.get('RADIANCERADIANCERADIANCE_ADD_BAND_10');
17 var MdB10 = single.get('RADIANCERADIANCERADIANCE_MULT_BAND_10');
18
19 print(AdB10);
20 print(MdB10);
21
22 var radB10 = single.select('B10').multiply(0.0003342).add(0.10000000149)
23 var greyscale = {min: 8, max: 11, palette: ['black', 'white']};
24 Map.addLayer(rdB10, greyscale, 'RAD B10');
25
26

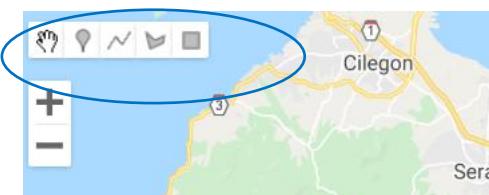
```

Simpanlah script anda dengan memilih tombol save lalu berinama misalkan 'KAL RADIOMETRIK'.

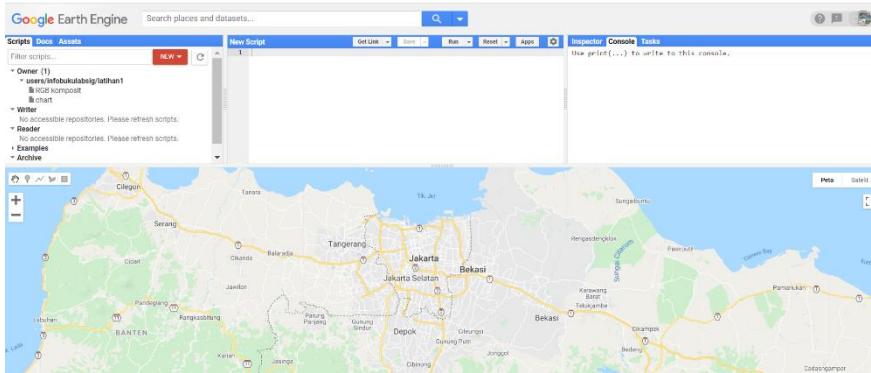


4.2. Memotong (Subset) dan mosaic citra (Studi kasus: Sentinel 2A)

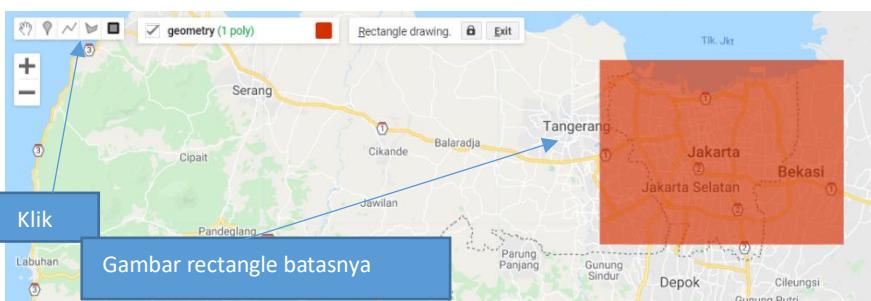
Dalam sesi ini, anda akan belajar untuk memotong (subset) citra. Untuk memotong citra, maka anda harus menyiapkan batas areanya terlebih dahulu. Batas area ini bisa anda gambar melalui tools geometry dimana anda dapat menggambar sendiri baik itu bentuk persegi (rectangle) atau poligon segi tak beraturan. Cara menggambar batas area ini sama seperti saat anda membuat training area pada sub-bab sebelumnya.



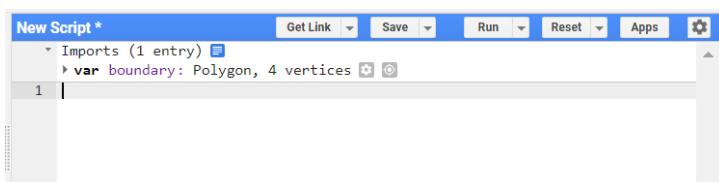
Mari kita mulai diawali dengan script kosong, lalu tambahkan dengan menggambar batas area. Batas area yang akan digambar ini akan menjadi pemotong citra. Selanjutnya, geser maps dan carilah area di sekitar DKI jakarta.



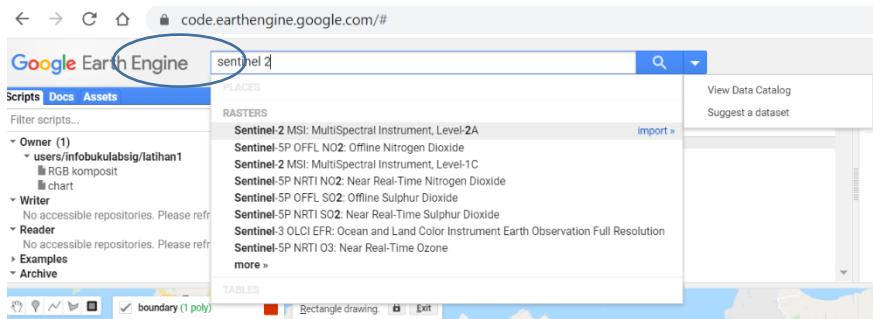
Selanjutnya buatlah poligon untuk menggambar batas areanya menggunakan *tools geometry*, anda dapat menggambar poligon persegi atau poligon segi tak beraturan . Sebagai contoh, buatlah poligon rectangle dengan area menutupi wilayah DKI Jakarta.



Setelah poligon tergambar, secara otomatis pada kolom new script di sisi atas, muncul variabel baru yakni var **geometry**. Gantilah nama (rename) variabel tersebut, misalkan menjadi var **boundary**.



Selanjutnya munculkan dataset satelitnya. Data satelit yang akan kita gunakan untuk latihan ini adalah data Sentinel 2-A. Mulailah untuk mencari dataset untuk data Sentinel 2-A melalui kolom search dengan mengetik "sentinel 2", lalu pilihlah **Sentinel-2 MSI: MultiSpectral Instrument, Level-2A**.



Selanjutnya catat kode *imageCollection ID* yakni **COPERNICUS/S2_SR**

Sentinel-2 MSI: MultiSpectral Instrument, Level-2A

Sentinel-2 is a wide-swath, high-resolution, multi-spectral imaging mission supporting Copernicus Land Monitoring studies, including the monitoring of vegetation, soil and water cover, as well as observation of inland waterways and coastal areas.

The Sentinel-2 L2 data are downloaded from scihub. They were computed by running sen2cor. WARNING: ESA did not produce L2 data for all L1 assets, and earlier L2 coverage is not global.

The assets contain 12 UINT16 spectral bands representing SR scaled by 10000 (unlike in L1 data, there is no B10). There are also several more L2-specific bands (see band list for details). See the [Sentinel-2 User Handbook](#) for details. In addition, three QA bands are present where one (QA60) is a bitmask band with cloud mask information. For more details, see the [full explanation of how cloud masks are computed](#).

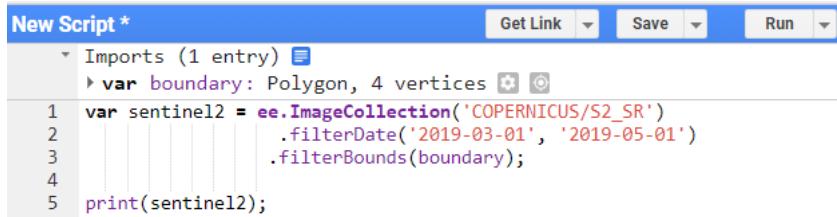
EE asset ids for Sentinel-2 L2 assets have the following format:
COPERNICUS/S2_SR/20151128T002653_20151128T102149_T56MNN. Here the first numeric part represents the sensing date and time, the second numeric part represents the product generation date and time, and the final 6-character string is

Selanjutnya buatlah script variabel baru yakni var **sentinel2**, dimana variabel ini berisi pencarian scenes sentinel 2 di area sekitar DKI jakarta sesuai batas area yang tergambar pada var **boundary**, lalu tanggal perekaman pada Bulan Maret hingga Mei tahun 2019. Berikut adalah script dari var **sentinel2** tersebut.

```
New Script * Get Link Save Run
Imports (1 entry)
var boundary: Polygon, 4 vertices
1 var sentinel2 = ee.ImageCollection('COPERNICUS/S2_SR')
2   .filterDate('2019-03-01', '2019-05-01')
3   .filterBounds(boundary);
4
```

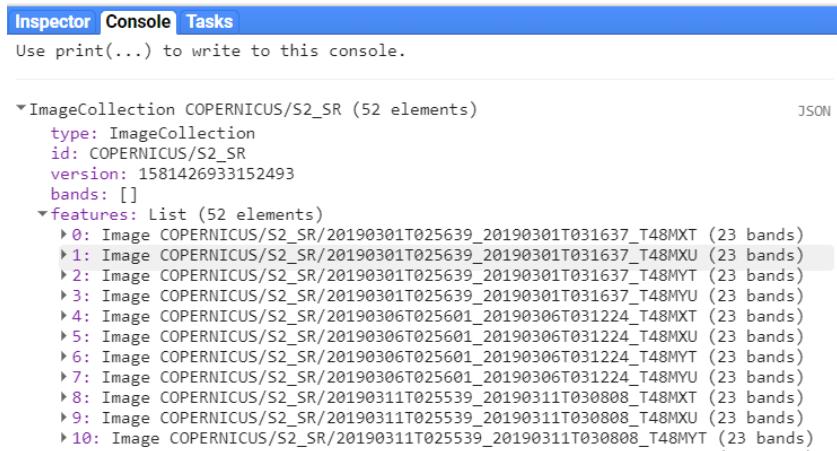
Script `.filterBounds (boundary)` mengartikan bahwa pencarian dataset hanya pada scenes yang mencover area seluas var **boundary**. Script ini bisa diisi baik geometry titik maupun poligon.

Selanjutnya gunakan script print() terhadap var **sentinel2**, untuk memunculkan list scenes yang dicari.



```
New Script *
Get Link Save Run
Imports (1 entry)
var boundary: Polygon, 4 vertices
1 var sentinel2 = ee.ImageCollection('COPERNICUS/S2_SR')
2   .filterDate('2019-03-01', '2019-05-01')
3   .filterBounds(boundary);
4
5 print(sentinel2);
```

Kemudian klik **run**, lalu cek lah pada kolom **Console**.



```
Inspector Console Tasks
Use print(...) to write to this console.

▼ImageCollection COPERNICUS/S2_SR (52 elements) JSON
  type: ImageCollection
  id: COPERNICUS/S2_SR
  version: 1581426933152493
  bands: []
  ▼features: List (52 elements)
    ▷ 0: Image COPERNICUS/S2_SR/20190301T025639_20190301T031637_T48MXT (23 bands)
    ▷ 1: Image COPERNICUS/S2_SR/20190301T025639_20190301T031637_T48MXU (23 bands)
    ▷ 2: Image COPERNICUS/S2_SR/20190301T025639_20190301T031637_T48MYT (23 bands)
    ▷ 3: Image COPERNICUS/S2_SR/20190301T025639_20190301T031637_T48MYU (23 bands)
    ▷ 4: Image COPERNICUS/S2_SR/20190306T025601_20190306T031224_T48MXT (23 bands)
    ▷ 5: Image COPERNICUS/S2_SR/20190306T025601_20190306T031224_T48MXU (23 bands)
    ▷ 6: Image COPERNICUS/S2_SR/20190306T025601_20190306T031224_T48MYT (23 bands)
    ▷ 7: Image COPERNICUS/S2_SR/20190306T025601_20190306T031224_T48MYU (23 bands)
    ▷ 8: Image COPERNICUS/S2_SR/20190311T025539_20190311T030808_T48MXT (23 bands)
    ▷ 9: Image COPERNICUS/S2_SR/20190311T025539_20190311T030808_T48MXU (23 bands)
    ▷ 10: Image COPERNICUS/S2_SR/20190311T025539_20190311T030808_T48MYT (23 bands)
```

Terlihat dari hasil pencarian dengan boundary (batas area) yang dibuat menghasilkan 52 scenes. Bila hasil scenes yang anda hasilkan jumlahnya berbeda bisa jadi penggambaran batas boundarynya berbeda, karena bila batas area yang digunakan lebih luas, bisa jadi akan muncul sejumlah scenes tambahan.

Selanjutnya pilihlah salah satu scenes dari list tersebut misalkan scenes id 'COPERNICUS/S2_SR/20190425T025551_20190425T031331_T48MXU', kemudian buatlah variabel baru misalkan dengan nama var **single_S2A**, gunakan script ee.Image() untuk memunculkan 1 scenes dengan id yang sudah dipilih.

Sebagaimana cara untuk memunculkan tampilan dalam komposit RGB yang sudah dibahas pada sub-bab sebelumnya, buatlah variabel baru misalkan var **RGBTrue** yang berisi hanya 3 band untuk menampilkan komposit RGB True Color. Sentinel 2A memiliki 12 band,

dimana Band 2, band 3 dan band 4 adalah band yang masuk pada spektrum tampak (VISIBLE). Sehingga untuk membuat tampilan RGB, maka susunannya adalah B4, B3, dan B2.

Sentinel-2 Bands	Central Wavelength (μm)	Resolution (m)
Band 1 - Coastal aerosol	0.443	60
Band 2 - Blue	0.490	10
Band 3 - Green	0.560	10
Band 4 - Red	0.665	10
Band 5 - Vegetation Red Edge	0.705	20
Band 6 - Vegetation Red Edge	0.740	20
Band 7 - Vegetation Red Edge	0.783	20
Band 8 - NIR	0.842	10
Band 8A - Vegetation Red Edge	0.865	20
Band 9 - Water vapour	0.945	60
Band 10 - SWIR - Cirrus	1.375	60
Band 11 - SWIR	1.610	20
Band 12 - SWIR	2.190	20

Selanjutnya buatlah variabel untuk parameter visualisasi citra, misalkan var **RGBparam**. Sentinel 2A memiliki resolusi radiometrik 12 bit, artinya nilai piksel berada pada rentang 0 – 2^{12} atau 0 – 4096. Pada parameter visual ini maka gunakan nilai minimum adalah 0 dan maksimum adalah 3000 agar tampilan citra lebih cerah.

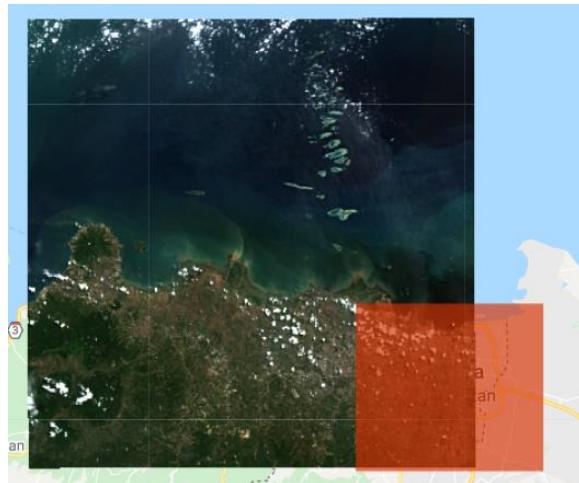
Terakhir tambahkan script Map.addLayer dengan memasukkan unsur var **RGBtrue** dan var **RGBparam**, berinama layer sebagai "RGB TCC"

```

New Script *
Get Link Save Run Reset Apps
Imports (1 entry)
var boundary: Polygon, 4 vertices
1 var sentinel2 = ee.ImageCollection('COPERNICUS/S2_SR')
2   .filterDate('2019-03-01', '2019-05-01')
3   .filterBounds(boundary);
4
5 print(sentinel2);
6
7 var single_S2A = ee.Image('COPERNICUS/S2_SR/20190425T025551_20190425T031331_T48MXU');
8
9 var RGBTrue = single_S2A.select(['B4', 'B3', 'B2']);
10 var RGBparam = { min: 0, max: 3000,};
11 Map.addLayer(RGBTrue, RGBparam, 'TRUE');

```

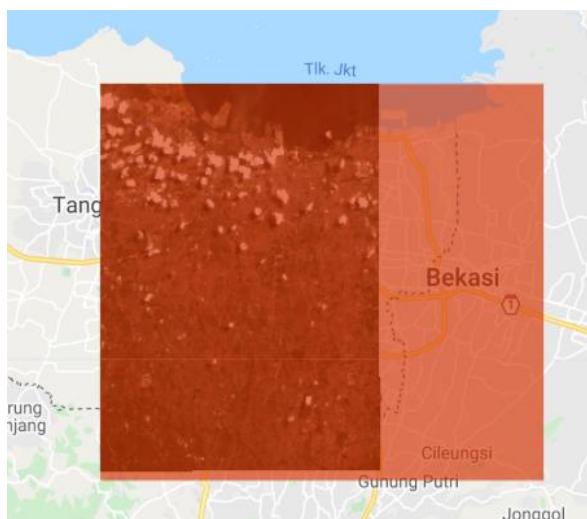
Klik tombol **Run**, maka akan muncul tampilan citranya dengan id 'COPERNICUS/S2_SR/20190425T025551_20190425T031331_T48MXU' dalam komposit RGB *True Color Composit*.



Terlihat scenes yang muncul areanya melebihi batas poligon boundary yang dibuat. Anda dapat memotongnya sesuai citra sesuai batas poligonnya. Pada var **RGBTrue**, tambahkan scrip `.clip()`. Script ini berfungsi untuk memotong citra sesuai batas area yang diinginkan. Tambahkan script `.clip()` dengan memasukkan var **boundary** didalamnya. Berikut scriptnya.

```
New Script *
Get Link Save Run Reset Apps
Imports (1 entry) ↗
  ↗ var boundary: Polygon, 4 vertices ↗
1 var sentinel2 = ee.ImageCollection('COPERNICUS/S2_SR')
2   .filterDate('2019-03-01', '2019-05-01')
3   .filterBounds(boundary);
4
5 print(sentinel2);
6
7 var single_S2A = ee.Image('COPERNICUS/S2_SR/20190425T025551_20190425T031331_T48MXU');
8
9 var RGBTrue= single_S2A.select(['B4', 'B3', 'B2'])
10   .clip(boundary);
11
12 var RGBparam = { min: 0, max: 3000,};
13 Map.addLayer(RGBTrue, RGBparam, 'TRUE');
```

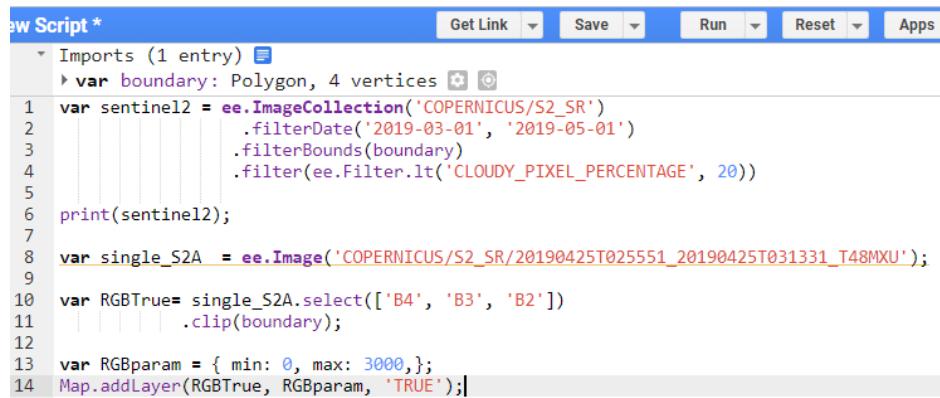
Klik tombol **run** untuk memunculkan



Terlihat pada scenes di atas, citra sentinel sudah terpotong, namun terdapat area yang kosong pada sisi kanan. Hal tersebut karena, scenes ID yang dipilih sebelumnya tidak mengcover sampai area tersebut, sehingga kita perlu menggabungkan dengan scenes id sebelahnya agar seluruh area yang terdapat didalam boundary terliput oleh citra sentinel.

Karena yang akan dimunculkan adalah gabungan beberapa scenes citra, maka script yang digunakan adalah `ee.ImageCollection` dan bukan `ee.image`. Maka variabel yang sudah menggunakan script `ee.ImageCollection` adalah var **sentinel2**, maka kita akan mengedit var sentinel 2 dengan menambahkan beberapa script filter.

Pada var **sentinel2**, tambahkan filter untuk mencari scenes dengan nilai persentase tutupan awan sesuai yang diinginkan. Misalkan kita ingin hanya scenes yang tutupan awannya kurang dari 20%, maka tambahkan script `.filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))`. Berikut scriptnya.

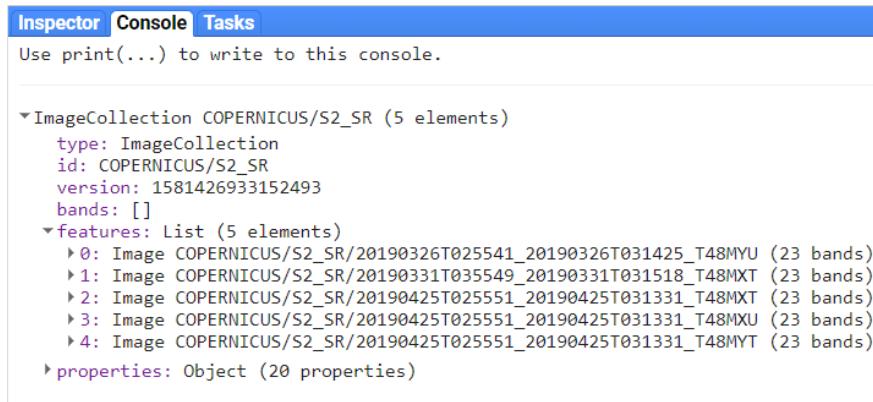


```

New Script *
Get Link Save Run Reset Apps
Imports (1 entry)
var boundary: Polygon, 4 vertices
var sentinel2 = ee.ImageCollection('COPERNICUS/S2_SR')
  .filterDate('2019-03-01', '2019-05-01')
  .filterBounds(boundary)
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
print(sentinel2);
var single_S2A = ee.Image('COPERNICUS/S2_SR/20190425T025551_20190425T031331_T48MXU');
var RGBTrue= single_S2A.select(['B4', 'B3', 'B2'])
  .clip(boundary);
var RGBparam = { min: 0, max: 3000,};
Map.addLayer(RGBTrue, RGBparam, 'TRUE');

```

Kemudian klik tombol **Run**, lalu cek lah pada kolom console.



```

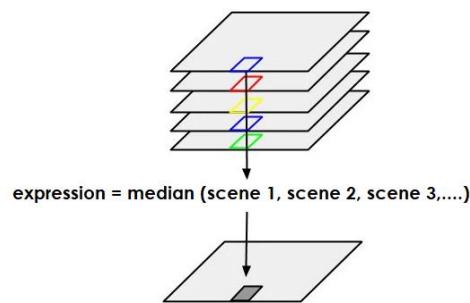
Inspector Console Tasks
Use print(...) to write to this console.

▼ImageCollection COPERNICUS/S2_SR (5 elements)
  type: ImageCollection
  id: COPERNICUS/S2_SR
  version: 1581426933152493
  bands: []
  features: List (5 elements)
    ▷ 0: Image COPERNICUS/S2_SR/20190326T025541_20190326T031425_T48MYU (23 bands)
    ▷ 1: Image COPERNICUS/S2_SR/20190331T035549_20190331T031518_T48MXT (23 bands)
    ▷ 2: Image COPERNICUS/S2_SR/20190425T025551_20190425T031331_T48MXT (23 bands)
    ▷ 3: Image COPERNICUS/S2_SR/20190425T025551_20190425T031331_T48MXU (23 bands)
    ▷ 4: Image COPERNICUS/S2_SR/20190425T025551_20190425T031331_T48MYT (23 bands)
  properties: Object (20 properties)

```

Terlihat, terjadi pengurangan sejumlah scenes, yang semula berjumlah 52 scenes berkurang menjadi 5 scene. Kelima scene tersebut adalah scene dengan kondisi persentase tutupan awan dibawah 20%.

Selanjutnya masih di var **sentinel2**, tambahkan script `.mean()`. Var **sentinel2** berisi 5 scenes, sehingga script `.median` ini digunakan untuk mereduksi dari kumpulan citra (imagecollection) tersebut yang kemudian diolah dimana scenes hasilnya memiliki nilai median dari seluruh scenes.



Kemudian script `.print()` serta var `single_S2A` tidak dibutuhkan lagi, anda bisa mematikan scriptnya dengan menambahkan garis miring 2 kali (//).

Kemudian var **RGBTrue** dilakukan modifikasi script, dimana **sentinel2** sebagai inputnya. Berikut script lengkapnya.

```

New Script * Get Link Save Run Reset Apps
Imports (1 entry)
  ↗ var boundary: Polygon, 4 vertices
1 var sentinel2 = ee.ImageCollection('COPERNICUS/S2_SR')
2   .filterDate('2019-03-01', '2019-05-01')
3   .filterBounds(boundary)
4   .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
5   .median();
6
7 //print(sentinel2);
8 //var single_S2A = ee.Image('COPERNICUS/S2_SR/20190425T025551_20190425T031331_T48MXU');
9
10 var RGBTrue= sentinel2.select(['B4', 'B3', 'B2'])
11   .clip(boundary);
12
13 var RGBparam = { min: 0, max: 3000,};
14 Map.addLayer(RGBTrue, RGBparam, 'TRUE');
15
  
```

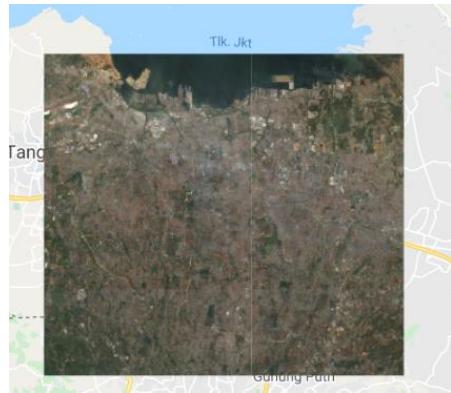
Klik run, hingga muncul tampilan citranya yang mengcover seluruh area boundary.



Bila terlihat masih banyak tutupan awannya, itu disebabkan karena pada perekaman bulan yang dipilih (Bulan Maret - Mei) tidak tersedia citra dengan scenes bebas awan. Selanjutnya anda dapat memperbesar rentang tanggal perekamannya, misalkan Bulan Maret hingga Desember untuk mendapatkan probabilitas hasil citra dengan awan yang lebih sedikit.

```
New Script * Get Link Save Run Reset Apps
Imports (1 entry)
  ↗ var boundary: Polygon, 4 vertices
1 var sentinel12 = ee.ImageCollection('COPERNICUS/S2_SR')
2   .filterDate('2019-03-01', '2019-12-01')
3   .filterBounds(boundary)
4   .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
5   .median();
6
7 //print(sentinel12);
8 //var single_S2A = ee.Image('COPERNICUS/S2_SR/20190425T025551_20190425T031331_T48MXU');
9
10 var RGBTrue= sentinel12.select(['B4', 'B3', 'B2'])
11   .clip(boundary);
12
13 var RGBparam = { min: 0, max: 3000,};
14 Map.addLayer(RGBTrue, RGBparam, 'TRUE');
```

Klik tombol **Run** kembali, maka akan terlihat tampilan citra dengan awan yang sudah berkurang.



Proses pemotongan dapat dilakukan dengan menggunakan data vektor yang kita punya. Misalkan kita memiliki data vektor batas administrasi dalam format shapefile. Kita dapat mengupload data shapefile kedalam asset earthengine, kemudian gunakan data tersebut untuk subset data citra. Masuk ke tab **Assets**, maka anda akan melihat bahwa list assets anda masih kosong. Lalu pilih **New**, kemudian klik pilih **Shapefiles**.

The screenshot shows the Google Earth Engine interface. The top navigation bar has tabs for 'Scripts', 'Docs', and 'Assets'. A red box highlights the 'Assets' tab, which is currently selected. Below it, a red button labeled 'NEW' is highlighted with a blue arrow pointing to it. To the right of the 'Assets' tab, a blue box contains the text 'Masuk Tab Assets'. On the left, a sidebar shows a user's folder structure: 'users/infobukulabsig'. A blue box labeled 'Klik New → Shapefiles' points to the 'New' button. The main workspace shows a code editor with a snippet for creating a shapefile from a polygon boundary. The code uses ee.ImageCollection('COPERNICUS/S2_SR') to filter for specific dates and then creates a polygon boundary. It then uses ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10) to filter for low cloud cover. Finally, it uses ee.FeatureCollection to create the shapefile and ee.Feature to add vertices. The code ends with a comment //var single_S2A = ee.Image('COPERNICUS/S2_SR/20190425T025551_2').

```
clip *  
  ▾ Imports (1 entry)  
    ▾ var boundary: Polygon, 4 vertices  
1 var sentinel2 = ee.ImageCollection('COPERNICUS/S2_SR')  
2   .filterDate('2019-03-01', '2019-12-01')  
3     .filterBounds(boundary)  
4       .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',  
5         10));  
6  
7           t(sentinel2);  
8 //var single_S2A = ee.Image('COPERNICUS/S2_SR/20190425T025551_2').  
9  
10 var RGBTrue= sentinel2.select(['B4', 'B3', 'B2'])  
11   .clip(boundary);
```

Kemudian akan muncul kotak dialog seperti dibawah, klik **select**.

Upload a new shapefile asset

Source files

SELECT Please drag and drop or select files for this asset.
Allowed extensions: shp, zip, dbf, prj, shx, cpg, fix, qix, sbn or shp.xml.

Asset ID
users/infobukulabsig/ Asset Name **Klik Select**

Properties

Metadata properties about the asset which can be edited during asset upload and after ingestion. The "system:time_start" property is used as the primary date of the asset.

Add start time Add end time Add property

Advanced options

Character encoding
UTF-8  

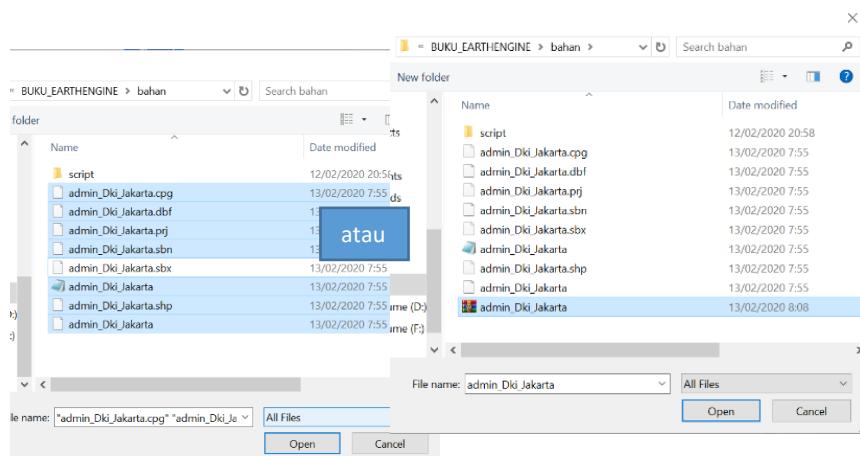
Maximum error
1.0 

Maximum vertices
1000000 

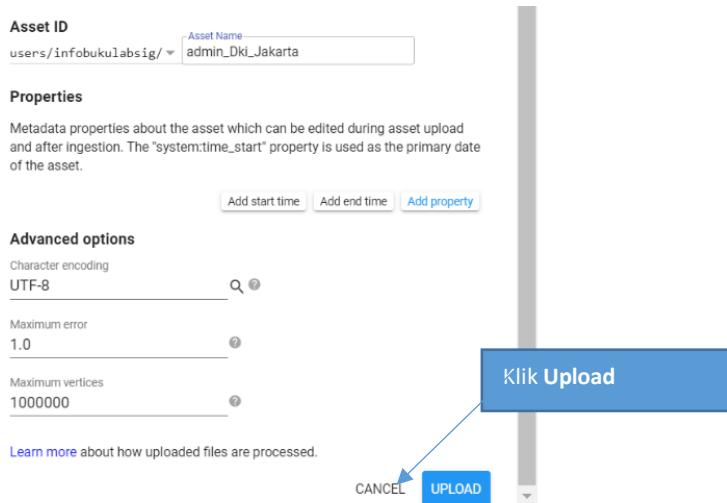
[Learn more](#) about how uploaded files are processed.

CANCEL UPLOAD

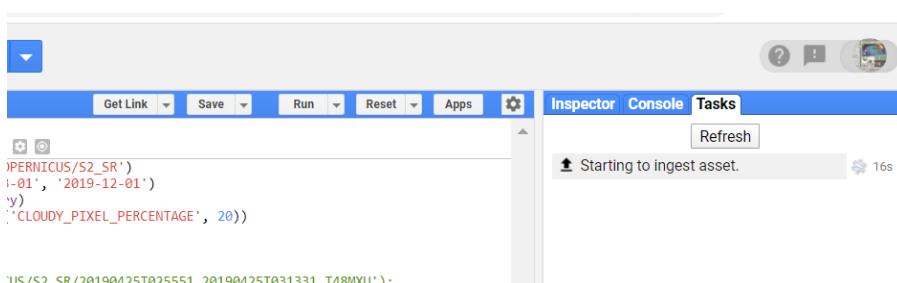
Kemudian carilah file shapefile yang mau anda upload. Data sampel tersedia didalam link yang diberikan, download dan pilihlah data *admin_DKI_Jakarta* yang sudah disediakan. Untuk mengupload, *select* seluruh file kecuali format *sbx*. Atau cara lain, anda dapat kompres seluruh file shapefile-nya kedalam zip/rar, lalu upload file zip/rar tersebut.



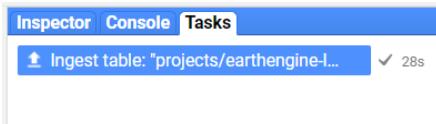
Kemudian klik **upload**



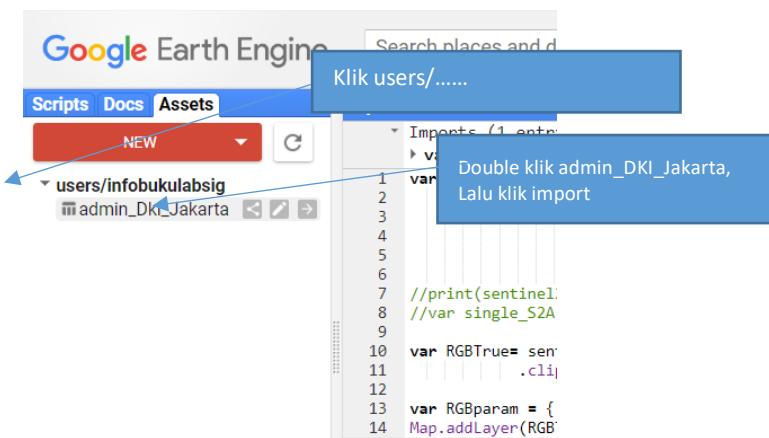
Cek pada tab *Tasks* untuk melihat progres upload. Semakin besar ukuran data yang anda upload, semakin lama proses upload data.



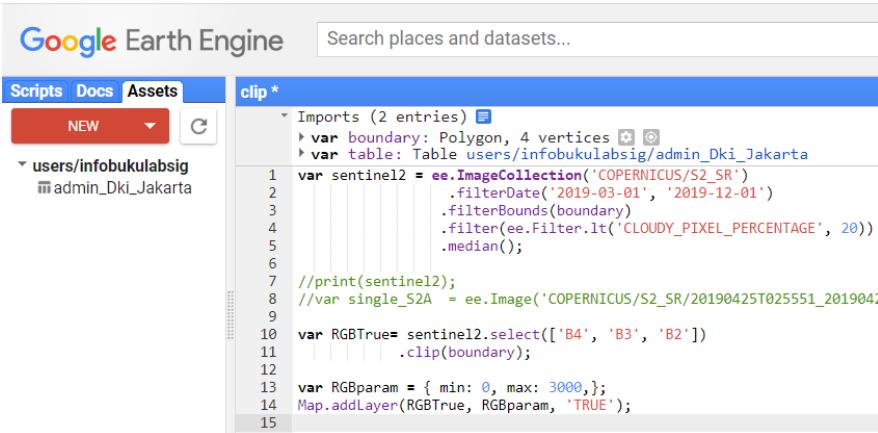
Berikut contoh proses upload telah berhasil.



Kemudian masuk ke Tab Assets kembali, klik tombol users/xxx maka anda akan melihat data admin_DKI_jakarta telah tersedia. Double klik admin_DKI_Jakarta tersebut, lalu pilih **import**.



Maka data tersebut masuk didalam code editor (sisi atas), dan menjadi variabel **table**.



Selanjutnya, gunakan variabel table ini yang berisi data poligon admin Dki Jakarta sebagai pemotong citra menggantikan var *boundary*. Editlah pada var **RGBTrue**, didalam script .clip() gantilah menjadinya "boundary" menjadi "table".

Berikut script nya.

```
p*
  ↴ Imports (2 entries) ⚒
    ↴ var boundary: Polygon, 4 vertices ⚒ ⓘ
    ↴ var table: Table users/infobukulabsig/admin_Dki_Jakarta
1  var sentinel12 = ee.ImageCollection('COPERNICUS/S2_SR')
2      .filterDate('2019-03-01', '2019-12-01')
3      .filterBounds(boundary)
4      .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
5      .median();
6
7 //print(sentinel12);
8 //var single_S2A = ee.Image('COPERNICUS/S2_SR/20190425T025551_20190425T031331_T48MXU');
9
10 var RGBTrue= sentinel12.select(['B4', 'B3', 'B2'])
11     .clip(table);
12
13 var RGBparam = { min: 0, max: 3000,};
14 Map.addLayer(RGBTrue, RGBparam, 'TRUE');
15
```

Klik tombol **Run**, dan anda akan mendapatkan citra terpotong sesuai batas administrasi DKI Jakarta.

Klasifikasi adalah teknik pengolahan pada citra dengan cara mengelompokkan piksel-piksel kedalam sejumlah kelas, sehingga setiap kelas memiliki pola-pola atau distribusi spasial yang unik dan spesifik yang mencerminkan suatu obyek atau informasi yang bermanfaat sesuai dengan keperluan (Chein-I Chang dan H. Ren, 2000).

Setiap piksel pada citra memiliki nilai atau yang disebut sebagai Digital Number (DN) atau nilai piksel (Pixel Value), dimana nilai tersebut merepresentasikan tingkat reflektan atau radiasi obyek di permukaan bumi. Setiap obyek di permukaan bumi tentu memiliki karakteristik pantulan maupun pancaran yang spesifik dan berbeda-beda. Dengan memahami karakteristik nilai piksel pada suatu obyek maka kita akan mendapatkan sebaran seluruh obyek tersebut di seluruh scenes citra. Piksel-piksel yang memiliki kriteria yang sama, akan tergabung dan membentuk satu kelas yang mencerminkan obyek tersebut.

Klasifikasi ini bertujuan untuk men-generalkan tampilan citra mentah baik greyscale maupun Tampilan RGB sehingga menghasilkan informasi spasial dengan tampilan yang mudah untuk diinterpretasi dan dipahami.

BAB 5 PENAJAMAN

Penajaman citra bertujuan untuk meningkatkan kualitas visual citra sehingga mempermudah user dalam proses interpretasi. Peningkatan kualitas visual citra dapat dilihat dari aspek spasial dan aspek spektral. Dalam aspek spasial, suatu citra dapat ditingkatkan resolusi spasialnya dengan melakukan proses pansharpen. Pansharpening adalah suatu metode menggabungkan (fusi) antara citra monochrome/panchromatic (hitam-putih) yang memiliki resolusi lebih tinggi dengan multispectral (berwarna) yang memiliki resolusi lebih rendah sehingga menghasilkan citra multispectral berwarna dengan resolusi yang lebih tinggi.

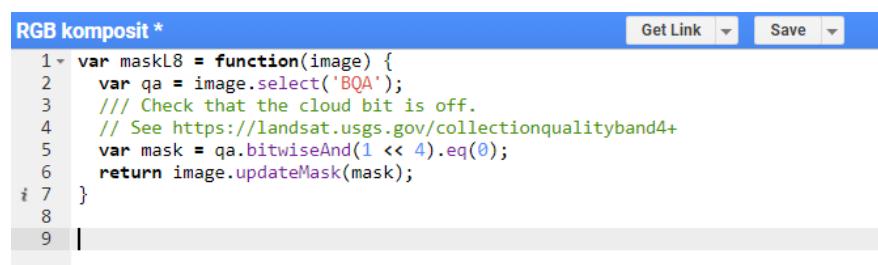
Sedangkan penajaman spektral atau biasa diistilahkan sebagai spectral enhancement, didefinisikan sebagai suatu proses memanipulasi kontras pada citra untuk meningkatkan kualitas visual sehingga informasi yang terdapat pada citra menjadi lebih mudah teridentifikasi.

5.1. Cloud mask (studi kasus: Landsat 8 TOA)

Sebagaimana diketahui, kelemahan penggunaan data optis adalah pada distribusi awan. Pada sub-bab sebelumnya, anda dapat menghasilkan citra yang merupakan hasil filter dengan mereduksi persentase tutupan awannya. Namun cara tersebut kurang optimal dalam mereduksi tutupan awan, karena hanya berlandaskan kepada filter scenes. Membersihkan awan dapat dilakukan yakni dengan menambal piksel berawan dengan citra beda perekaman yang tidak berawan. Proses ini dapat dilakukan dengan memanfaatkan layer BQA yakni band berisi penilaian kualitas citra. BQA merupakan band tersendiri, dimana image ini berisi informasi yang bisa menunjukkan piksel-piksel berawan, bayangan awan, saturasi radiometric, dll. Melalui layer BQA, piksel-piksel bermasalah tersebut dihapus (masking) dan diganti dengan piksel pada tanggal perekaman lain. Sehingga hasil akhir akan menghasilkan citra hasil tambal sulam sehingga menghasilkan kualitas citra yang lebih baik.

Buatlah script baru, dan hapus script lama (sub-bab) sebelumnya. Berikut ini adalah script untuk mereduksi piksel awan. Script ini memanfaatkan layer BQA pada Landsat 8 dimana layer tersebut berisikan informasi mengenai piksel awan yakni piksel dengan nilai 1 sampai 4. Sehingga piksel-piksel awa tersebut dimasking menjadi nilai 0 dan diisi kembali dengan piksel non awan.

Fungsi ini didefinisikan dalam suatu variable fungsi, yaitu var **maskL8**.



```
RGB komposit * Get Link Save
1 var maskL8 = function(image) {
2   var qa = image.select('BQA');
3   /// Check that the cloud bit is off.
4   // See https://ladsat.usgs.gov/collectionqualityband4+
5   var mask = qa.bitwiseAnd(1 << 4).eq(0);
6   return image.updateMask(mask);
7 }
8
9 |
```

Langkah selanjutnya anda dapat membuat boundary (batas area) dalam bentuk polygon menggunakan tools geometry.



Setelah anda menggambar polygon boundary, maka akan muncul var geometry pada sisi kolom *code editor*.

Selanjutnya buatlah var datanya, yakni kita mencoba mengambil data Landsat 8 yang sudah terkoreksi Top of Atmosferik (TOA). Data tersebut memiliki ID 'LANDSAT/LC08/C01/T1_TOA'. Kemudian gunakan ".filterDate" untuk menfilter tanggal perekaman antara misalkan 2017 hingga 2019. Lalu ".filterBounds" untuk menfilter dari sisi lokasi pencarian citra.

Kemudian .map() diisikan dengan fungsi *masking cloud* yang sebelumnya sudah terdefinisi dalam variable **maskL8**. Fungsi ini nanti yang akan mereplace piksel berawan dengan piksel non-awan pada rentang periode 2017 dan 2019 sesuai .filterDate yang sudah ditentukan. Kemudian script .median merupakan script yang berfungsi untuk mereduksi dari kumpulan citra (imagecollection) yang kemudian diolah menjadi 1 nilai rerata yakni menerapkan fungsi median.

Berikut adalah script lengkapnya:

```
RGB komposit *
Get Link Save
Imports (1 entry)
var geometry: Polygon, 4 vertices
var maskL8 = function(image) {
  var qa = image.select('BQA');
  // Check that the cloud bit is off.
  // See https://landsat.usgs.gov/collectionqualityband4+
  var mask = qa.bitwiseAnd(1 << 4).eq(0);
  return image.updateMask(mask);
}
var composite = ee.ImageCollection('LANDSAT/LC08/C01/T1_TOA')
  .filterDate('2017-01-01', '2019-12-31')
  .filterBounds(geometry)
  .map(maskL8)
  .median()
```

Selanjutnya tambahkan var baru, yakni var **RGBTrue**, yang merupakan turunan dari var **composite** namun hanya terseleksi band tertentu, yakni band 4, band 3, dan band 2. Pemilihan 3 band tersebut dimaksudkan agar image dapat tervisualisasikan dalam komposit RGB yang mana membutuhkan (hanya) 3 band.

Lalu tambahkan var **visualisasi**, tipe parameter ini tentu adalah parameter visualisasi RGB (bukan greyscale). Sehingga hanya membutuhkan nilai min dan max. Karena data landsat 8 yang digunakan adalah data yang telah terkalibrasi TOA, maka rentang nilai piksel berbeda dengan data Landsat 8 tipe RAW. Data Landsat 8 TOA ini memiliki rentang nilai 0 sampai 0.5.

Selanjutnya tambahkan script *Map.addLayer*, dimana var **RGBTrue** sebagai input datasetnya, lalu var **RGBparam** sebagai parameter visualisasinya.

```
▼ Imports (1 entry) □
  ▶ var geometry: Polygon, 4 vertices ⚙ ⓘ
1 ▷ var maskL8 = function(image) {
2   var qa = image.select('BQA');
3   // Check that the cloud bit is off.
4   // See https://landsat.usgs.gov/collectionqualityband4+
5   var mask = qa.bitwiseAnd(1 << 4).eq(0);
6   return image.updateMask(mask);
7 }
8
9 var composite = ee.ImageCollection('LANDSAT/LC08/C01/T1_TOA')
10   .filterDate('2017-01-01', '2019-12-31')
11   .filterBounds(geometry)
12   .map(maskL8)
13   .median()
14
15 //visualisasi citra|
16 var RGBTrue = composite.select(['B4', 'B3', 'B2']);
17 var RGBparam = { min: 0, max: 0.3,};
18 Map.addLayer(RGBTrue, RGBparam, 'TRUE');
```

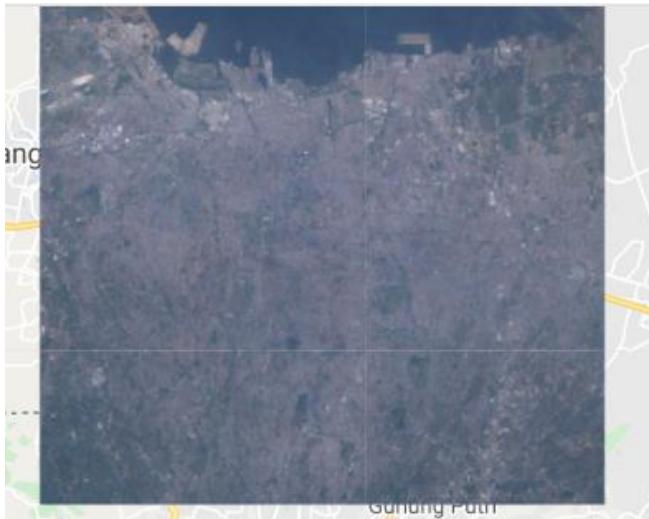
Klik **run** untuk memunculkan/visualisasikan citranya.



Anda dapat melakukan proses pemotongan (clipping) citra sesuai batas area (geometry). Lakukan editing pada var **RGBTrue**, dengan menambahkan script *.clip(geometry)*.

Kemudian klik tombol **Run** kembali, maka anda akan dapatkan hasil citra landsat 8 Jakarta bebas awan.

```
New Script * Get Link Save Run
  * Imports (1 entry) ↕
    ↗ var geometry: Polygon, 4 vertices ↗ ↗
1 var maskL8 = function(image) {
2   var qa = image.select('BQA');
3   // Check that the cloud bit is off.
4   // See https://landsat.usgs.gov/collectionqualityband4+
5   var mask = qa.bitwiseAnd(1 << 4).eq(0);
6   return image.updateMask(mask);
7 }
8
9 var composite = ee.ImageCollection('LANDSAT/LC08/C01/T1_TOA')
10   .filterDate('2017-01-01', '2019-12-31')
11   .filterBounds(geometry)
12   .map(maskL8)
13   .median()
14
15 //visualisasi citra
16 var RGBTrue = composite.select(['B4', 'B3', 'B2'])
17   .clip(geometry);
18 var RGBparam = { min: 0, max: 0.3,};
19 Map.addLayer(RGBTrue, RGBparam, 'TRUE');
```



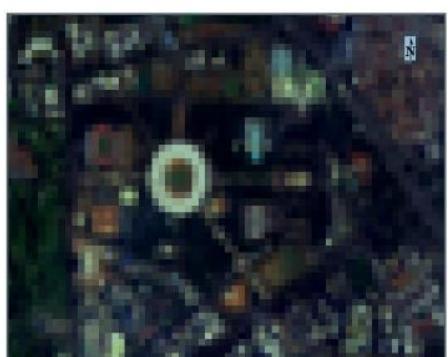
5.2. Pansharpening

Faktor resolusi atau kedetailan citra adalah kunci dalam proses interpretasi visual. Tentu, usaha interpretasi atau mengenali dan mengidentifikasi suatu obyek akan menjadi lebih mudah bila citra yang tersedia memiliki resolusi yang lebih tinggi. Saat ini, beberapa satelit telah menyediakan citra resolusi tinggi namun tidak selalu dalam mode multispectral atau

berwarna. Beberapa satelit memiliki band monochrome dan multispektral, contohnya antara lain:

- ALOS dimana sensor PRISM menyediakan citra resolusi 2,5 meter dengan mode pankromatik (hitam-putih). Sedangkan sensor AVNIR memiliki citra resolusi 10 m dengan mode multispektral 4 band (berwarna).
- SPOT 6 dimana memiliki 1 band pankromatik (hitam-putih) resolusi 1,5 meter dan 4 band multispektral (berwarna) resolusi 6 meter.
- Geo-eye 1 dimana memiliki 1 band pankromatik (hitam-putih) resolusi 0,41 meter dan 4 band multispektral (berwarna) resolusi 1,65 meter.
- Dan lain-lain.

Data Landsat pun termasuk pada jenis satelit yang menyediakan data pankromatik dan multispektral. Pada Landsat 8, band pankromatik terdapat pada band 8 memiliki resolusi 15 meter, lebih tinggi dibandingkan band multispektral lainnya yang memiliki resolusi 30 m. Berikut perbedaan tingkat kedetailan antara citra multispektral (resolusi 30 meter) yang telah dikompositkan menjadi citra berwarna dengan band 8 (resolusi 15 meter).



Band 1 hingga band 7



Band 8

Agar mendapatkan citra multispektral (berwarna) dengan resolusi yang lebih tinggi, maka perlu melakukan proses pansharpening. Proses ini yakni menggabungkan citra resolusi rendah berwarna dengan citra pankromatik resolusi tinggi. Metode fusi ini dikembangkan oleh Dr. Yun Zang dari Departemen Geodesi dan Geomatika Universitas New Brunswick. Proses fusi

ini dapat juga diterapkan pada citra beda sensor, misalkan antara SPOT 6 dengan Geo-eye, namun ada beberapa syarat yang harus diperhatikan, yakni:

1. Kedua citra yang akan di fuzi (multispektral dan pankromatik) harus memiliki rasio resolusi sebesar 5:1. Maksudnya adalah bila ingin melakukan panshaping Landsat multispektral dengan resolusi 30 meter, maka resolusi citra pankromatik yang masih diperbolehkan tidak boleh lebih kecil dari 6 meter.
2. Kedua citra yang akan di fuzi (multispektral dan pankromatik) harus memiliki geometri yang sama. Oleh karena itu, biasanya kedua citra beda sensor yang akan difusi harus melalui tahap koreksi geometric terlebih dahulu.
3. Tanggal perekaman kedua citra yang akan di fuzi haruslah dalam tempo yang berdekatan. Hal ini dimaksudkan agar potensi perubahan obyek yang terdapat didalam scenes citra tidak terlalu besar.

Proses melakukan pansharpening dengan Teknik yang sederhana yakni menggunakan Teknik HSV Sharpening. HSV merupakan kependekan dari Hue, Saturation, dan Value dimana karakteristik pokok dari warna tersebut adalah:

- Hue: menyatakan warna sebenarnya, seperti merah, violet, dan kuning dan digunakan menentukan kemerahannya (redness), kehijauannya (greeness), dsb.
- Saturation: kadang disebut chroma, adalah kemurnian atau kekuatan warna.
- Value: kecerahan dari warna. Nilainya berkisar antara 0-100 %. Apabila nilainya 0 maka warnanya akan menjadi hitam, semakin besar nilai maka semakin cerah dan muncul variasi-variasi baru dari warna tersebut.

Teknik penajaman ini diawali dengan transformasi RGB kedalam kombinasi HRV. Kemudian, unsur Value (V) diisi dengan layer resolusi lebih tinggi (pankromatik) untuk meresampling unsur H dan S kedalam resolusi yang sama dengan menggunakan teknik nearest neighbor, bilinear, atau cubic convolution. Ketiga layer tersebut masing-masing ditumpang-susun (*layerstacking*) menjadi layer RGB baru dengan komposisi Hue, Saturation, dan Pankromatik. Layer RGB baru ini tidak merubah warna dan tone namun sudah meningkat dari sisi resolusi spasial mengikuti resolusi dari layer pankromatik.

Pada sesi ini, anda akan belajar membuat citra komposit berwarna dari data Landsat 8 dengan resolusi 15 meter. Tahapan yang dilakukan adalah pertama data Landsat 8 TOA yang akan digunakan baiknya sudah diolah diterapkan cloud mask. Oleh karena itu, kita dapat gunakan script pada sub-bab sebelumnya dengan melanjutkan fungsi pansharpening.

Buatlah var baru yakni var **panSharpenL8** yang berisi fungsi:

- Definisikan var **rgb** sebagai layer hanya berisi Band 432.
- Definsikan var **pan** hanya Band 8.
- Definisikan var **Huesat** yang berisi fungsi transformasi RGBtoHRV, lalu select hanya Hue dan Sat.
- Definsikan var **upres** yang berisi fungsi penggabungan var **Huesat** dan var **pan**.

Sisipkan saja grup script dari var **panSharpenL8** misalkan diatas var **composite**. Kemudian pada komponen var composite, sisipkan .map() lalu inputkan var **panSharpenL8** didalamnya. Hal ini bertujuan data dari var **composite** menerapkan fungsi HRV-Pansharpen, sehingga outputnya hanya bertambah sebanyak 3 band saja yakni Red, Green, dan Blue. Sehingga, total dalam var composite terdapat 6 band, yakni 3 band dengan resolusi 30 meter (original) dengan kode layer "B4, B3, dan B2" dan 3 band lagi hasil HRV-Pansharpen dengan resolusi 15 meter dengan kode layer "red, green, dan blue".

Berikut adalah full scriptnya.

New Script *

Get Link Save Run

```

Imports (1 entry) ↗
↳ var geometry: Polygon, 4 vertices ⚙️ ⓘ

1 var maskL8 = function(image) {
2   var qa = image.select('BQA');
3   // Check that the cloud bit is off.
4   // See https://landsat.usgs.gov/collectionqualityband4+
5   var mask = qa.bitwiseAnd(1 << 4).eq(0);
6   return image.updateMask(mask);
7 }
8
9
10 var panSharpenL8 = function(image) {
11   var rgb = image.select('B4', 'B3', 'B2');
12   var pan = image.select('B8');
13   var huesat = rgb.rgbToHsv().select('hue', 'saturation');
14   var upres = ee.Image.cat(huesat, pan).hsvToRgb();
15   return image.addBands(upres);
16 };
17
18
19 var composite = ee.ImageCollection('LANDSAT/LC08/C01/T1_TOA')
20   .filterDate('2017-01-01', '2019-12-31')
21   .filterBounds(geometry)
22   .map(maskL8)
23   .map(panSharpenL8)
24   .median()
25

```

Tambahkan var **PAN** dimana var composite sebagai inputnya serta seleksi layernya dengan kode “red, green, blue” agar layer yang terseleksi adalah layer hasil HRV-pansharpen.

Kemudian tambahkan scrip Map.addLayer, inputkan var PAN, kemudian tambahkan var RGBparam sebagai parameter visualnya, dan berinama layer “pan”. Klik tombol Run, hingga pada map display muncul 2 layer yakni “pan” dan “TRUE”.

BAB 6 KLASIFIKASI CITRA

6.1. Supervised (Studi Kasus Data Landsat 8)

Menurut Lillesand dan Kiefer (1990), klasifikasi dibagi ke dalam dua jenis teknik yaitu.

a) Klasifikasi Supervised (Terbimbing)

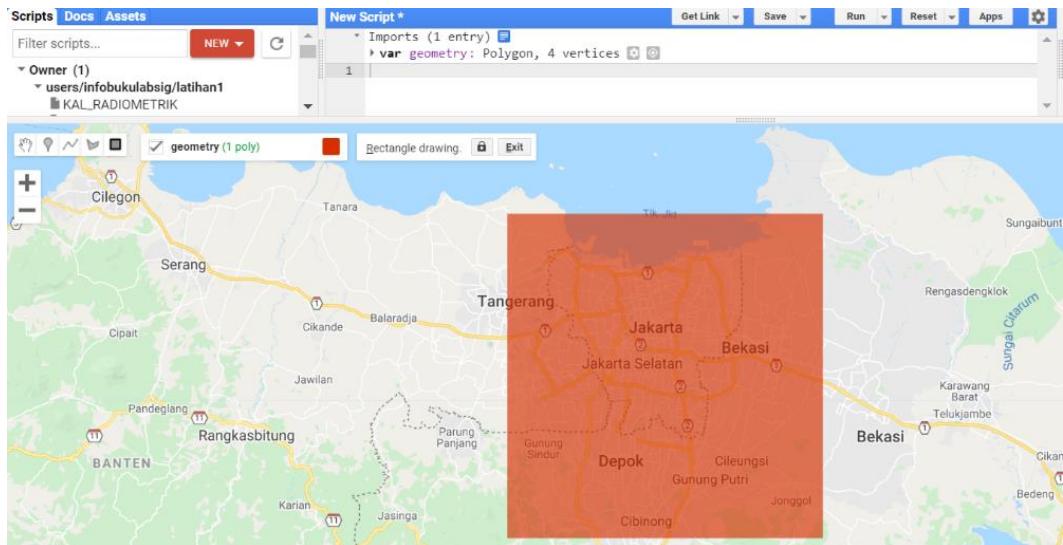
Teknik ini dilakukan dengan prosedur pengambilan sampel beberapa piksel untuk masing-masing kelas/obyek. Sampel atau Region Of Interest ini digunakan untuk mendapatkan karakteristik nilai piksel di masing-masing obyek/kelas. Kemudian seluruh piksel yang bukan sebagai sampel akan dikelompokkan dengan mengacu pada karakteristik nilai piksel sampel yang telah diambil dengan menerapkan perhitungan statistik.

b) Klasifikasi Unsupervised (Tidak Terbimbing)

Pada klasifikasi tidak terbimbing, pengklasifikasian dimulai dengan pemeriksaan statistik seluruh piksel dan membaginya kedalam kelas-kelas yang jumlahnya telah ditentukan. Dalam teknik ini, piksel dikelompokkan bukan atas dasar pengambilan sampel, namun atas dasar perhitungan statistik citra menggunakan algoritma klusterisasi. Dalam metode ini diawali dengan penentuan jumlah kelas (cluster) yang akan dibuat. Kemudian setelah kelas-kelas tersebut dihasilkan, lalu didefinisikan sesuai dengan obyek yang ingin diidentifikasi.

Pada Sesi ini kita akan belajar melakukan klasifikasi supervised untuk menghasilkan peta tutupan lahan. Data yang digunakan adalah data Landsat 8 dengan produk yang sudah terkalibrasi atau level TOA. Agar hasil tutupan lahan memiliki hasil yang baik, maka dataset perlu dilakukan preprocessing melalui tahap pembersihan awan.

Oleh karena itu, tahap awal batasi wilayah kerjanya misalkan untuk area sekitar DKI Jakarta. Gunakan alat penggambar geometri (Geometry Tools) untuk menggambar batas wilayah dalam bentuk persegi (rectangle).



Kemudian, tambahkan varibel masking seperti yang sudah pernah dijelaskan di subbab 5.1, dimana variabel ini menggunakan layer BQA pada Landsat 8. Buatlah variabel baru, misalkan var maskL8, sesuai script di bawah ini.

```
New Script * Get
Imports (1 entry)
var geometry: Polygon, 4 vertices

1 var maskL8 = function(image) {
2     var qa = image.select('BQA');
3     var mask = qa.bitwiseAnd(1 << 4).eq(0);
4     return image.updateMask(mask);
5 }
6
```

Selanjutnya, tambahkan variabel dataset yakni Landsat 8 TOA dengan periode perekaman misalkan 2018 hingga 2019. Kemudian inputkan var maskL8 dalam fungsi ".map" agar fungsi *cloud mask* berjalan. Selanjutnya gunakan fungsi ".median" dan ".clip". Selanjutnya tambahkan var RGBtrue dan var RGBparam untuk bisa memunculkan visualisasi citra. Berikut adalah script lengkapnya.

Klik **run** untuk mengecek visualisasi dataset yang sudah disesuaikan dengan filter dan parameter yang digunakan.

New Script *

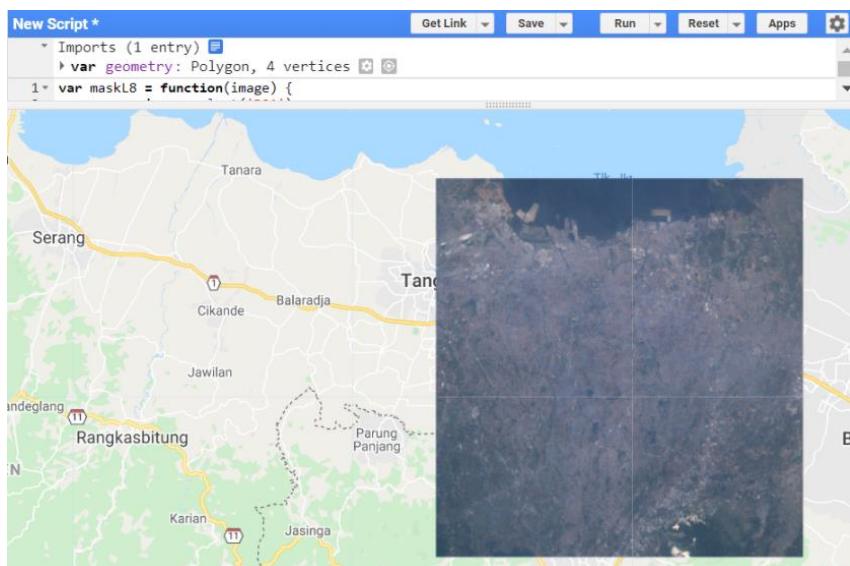
Get Link Save

```

Imports (1 entry) ↗
▶ var geometry: Polygon, 4 vertices ⚙️ ⓘ

1 var maskL8 = function(image) {
2   var qa = image.select('BQA');
3   var mask = qa.bitwiseAnd(1 << 4).eq(0);
4   return image.updateMask(mask);
5 }
6
7 var composite = ee.ImageCollection('LANDSAT/LC08/C01/T1_TOA')
8   .filterDate('2018-01-01', '2019-12-31')
9   .map(maskL8)
10  .median()
11  .clip(geometry)
12
13 var RGBTrue = composite.select(['B4', 'B3', 'B2']);
14 var RGBparam = { min: 0, max: 0.3,};
15 Map.addLayer(RGBTrue, RGBparam, 'TRUE');|
```

Maka akan muncul hasil visualisasi citra yang akan diklasifikasikan kedalam kelas tutupan lahan. Pastikan citra yang dijadikan sumber data ini tidak memiliki tutupan awan yang banyak. Bila tutupan awan masih tergolong banyak, anda bisa memperpanjang rentang periode perekamannya.



Langkah selanjutnya adalah membuat sampel atau training area. Dalam tata cara penggunaan metode klasifikasi supervised, anda harus mengecek kelas-kelas tutupan lahan yang ada di wilayah kajian terlebih dahulu. Anda harus melakukan observasi terlebih dahulu melalui visualisasi citra untuk memperkirakan obyek-obyek tutupan lahan apa saja yang ada di dalam liputan citra tersebut. Buatlah list untuk kelas-kelas tutupan lahan yang berhasil anda kenali.

Sistem penentuan kelas tutupan lahan sebaiknya berpedoman pada standar nasional nomor 8 tahun 2007. Dalam pedoman SNI tersebut, kelas-kelas tutupan lahan dibedakan

berdasarkan skala data citra yang digunakan. Standar tersebut berisi klasifikasi penutup lahan pada skala 1:1.000.000, 1:250.000, 1:50.000, dan 1:25.000. Semakin detail skala peta yang dihasilkan, maka semakin rinci dan kompleks kelas-kelas tutupan lahananya. Contohnya adalah klasifikasi obyek sawah yang ditampilkan pada kempat skala yang berbeda seperti berikut:

Tabel 6.1. Kedekatan Klasifikasi Penutup Lahan Berdasarkan Skala Peta

1:1.000.000	1.250.000	1:50.000	1:25.000
Sawah	Sawah	Sawah Irigsi	Sawah Irigsi
		Sawah Tadah Hujan	Sawah Tadah Hujan
	Sawah Pasang Surut	Sawah Lebak	Sawah Lebak
		Sawah Pasang Surut	Sawah Pasang Surut

Penentuan skala pada data citra dapat menggunakan rumus/aturan Aturan Tobler. Aturan ini menekankan pada ukuran obyek terkecil yang mampu ditampilkan dalam peta berdasarkan ukuran resolusi spasial atau ukuran piksel citra yang digunakan.

$$\text{Skala peta} = \text{Resolusi spasial citra (dalam meter)} * 2 * 1000$$

Bila kita menggunakan citra Landsat, resolusi 30 meter, maka sepadan dengan skala 1:60.000.

Selanjutnya, buatlah kelas-kelas tutupan lahan menggunakan geometry tools. Masukkan seluruh list kelas tutupan lahan kedalam geometry tools, dimana tipenya adalah *feature collection* dengan menambahkan keterangan "lc" sebagai id dalam angka untuk membedakan antara kelas tutupan lahan.

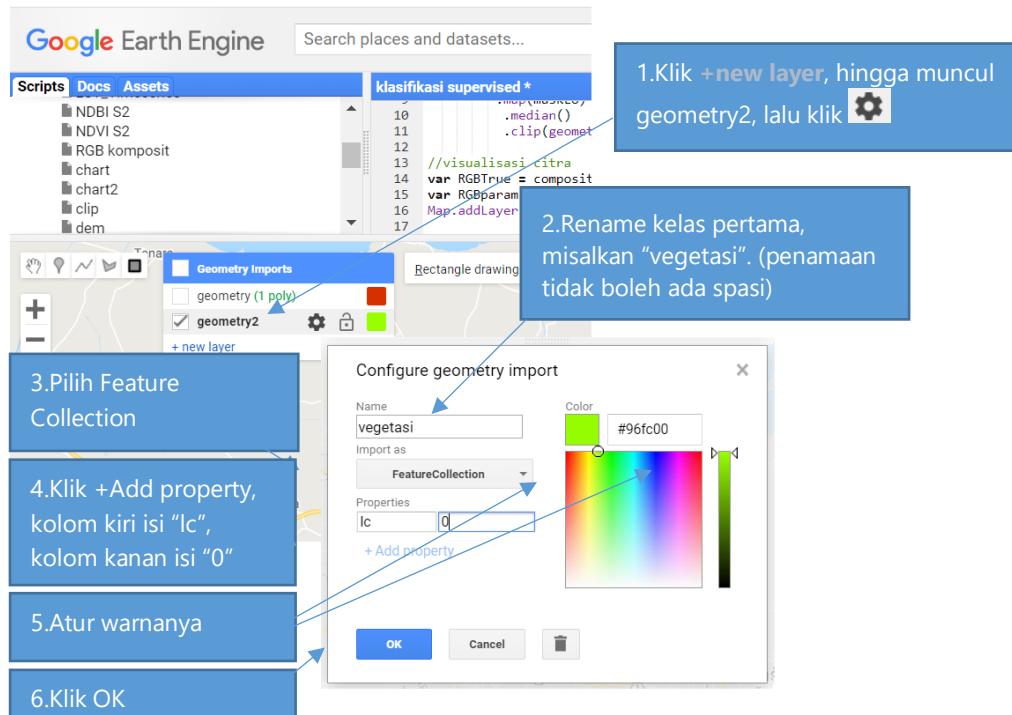
Misalkan daftar kelas tutupan lahan beserta tipe parameternya adalah sebagai berikut:

Tabel 6.2. Kelas Tutupan Lahan pada Earth ENgine

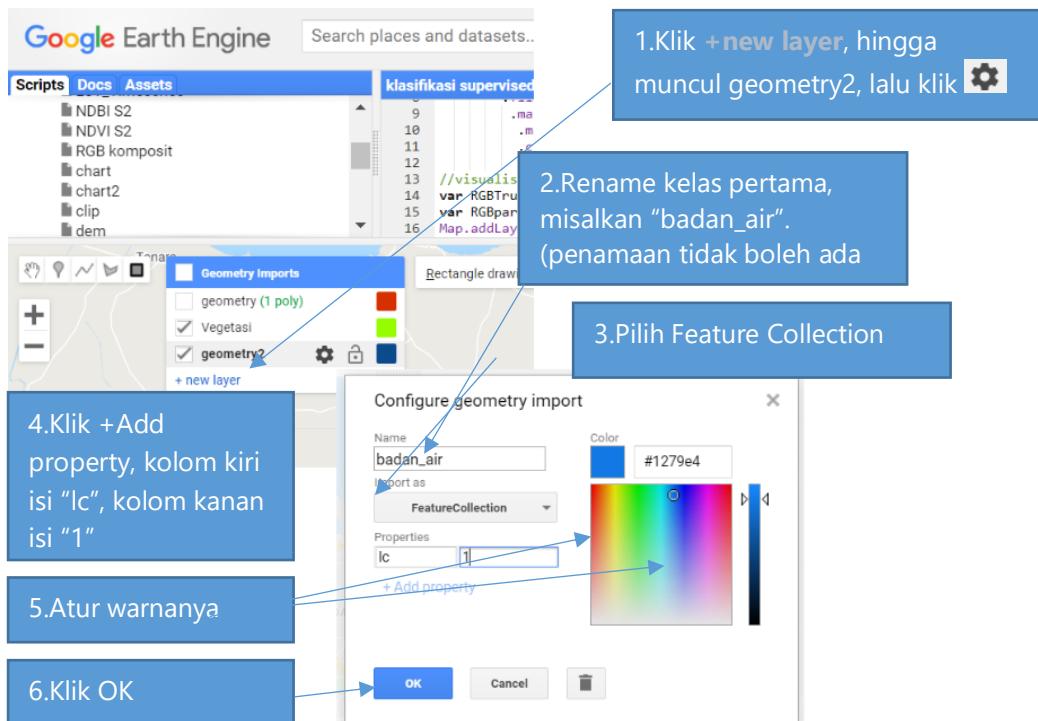
No	Kelas tutupan lahan	Type	lc
1	Vegetasi	Feature collection	0

2	Badan air	Feature collection	1
3	Lahan terbangun	Feature collection	2
4	Tambak	Feature collection	3
5	Rumput/Semak	Feature collection	4
6	Tanah Terbuka	Feature collection	5
7	Sawah	Feature collection	6

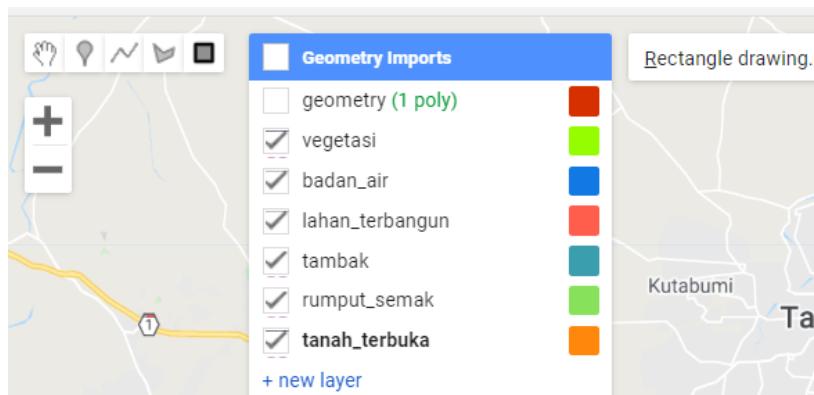
Berikut cara membuat sampel kelas-kelas tutupan lahan.



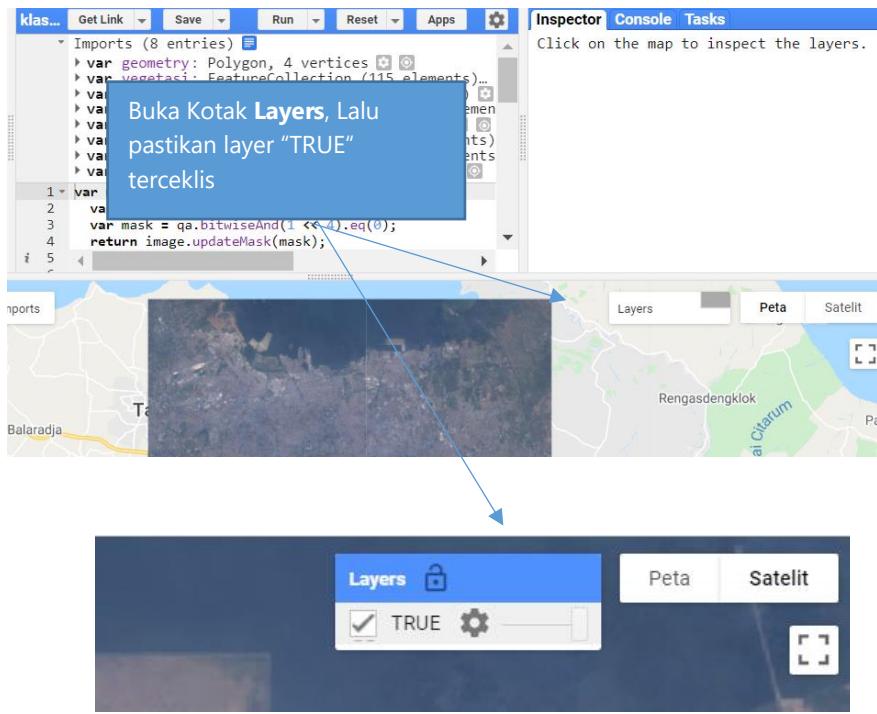
Selanjutnya buatlah kelas tutupan berikutnya, misalkan "badan_air" (penulisan tidak boleh menggunakan spasi, gantilah dengan "underscore"). Berikut langkahnya:



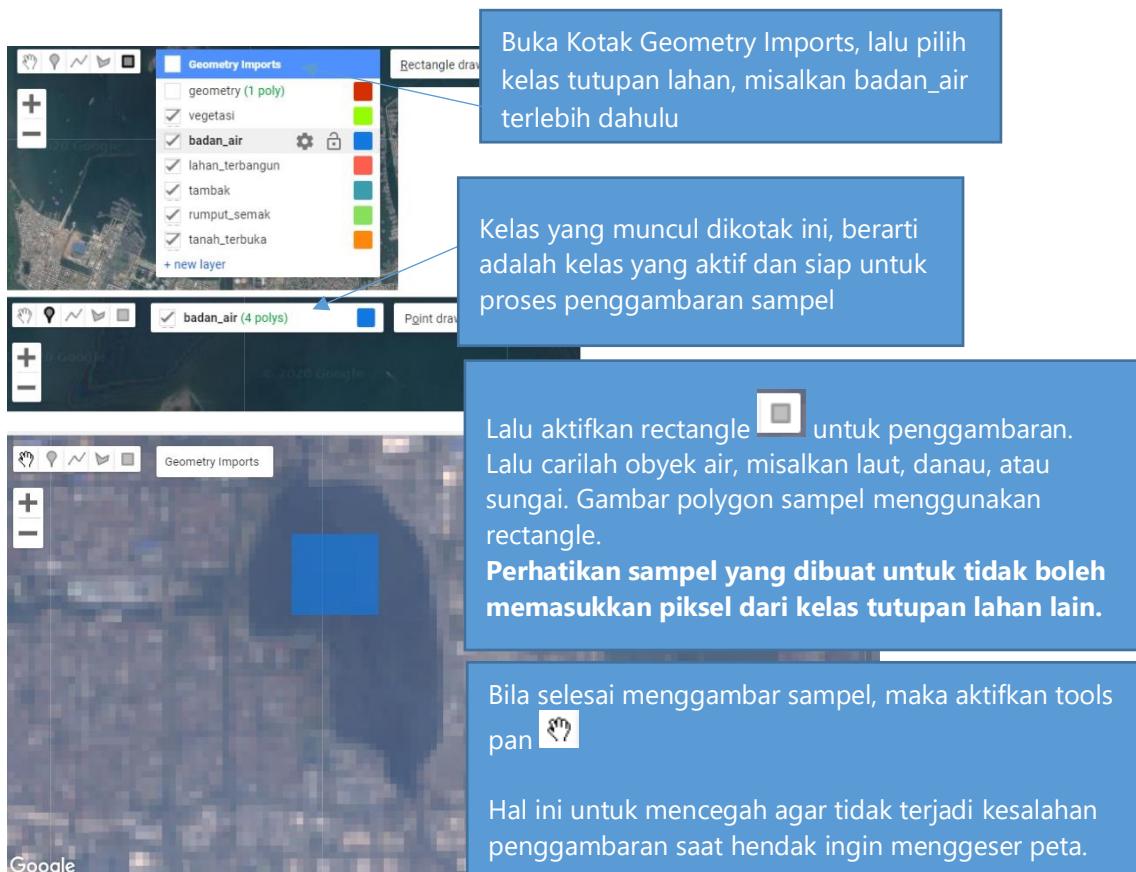
Lanjutkan hingga seluruh kelas tutupan lahan terbentuk didalam kolom Geometry Imports seperti gambar di bawah ini.



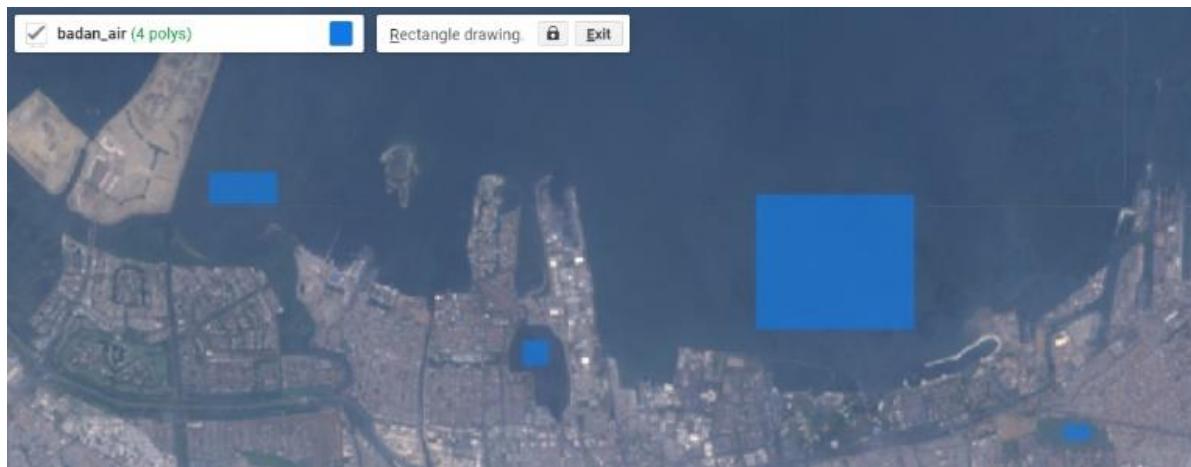
Langkah selanjutnya adalah menggambar sampel setiap kelas tutupan lahan. Anda harus mencari sampel piksel dari setiap kelas tutupan lahan. Sebagai contoh untuk kelas badan air, maka anda harus mencari sampel area di beberapa obyek seperti laut, sungai besar, dan danau. Pengamatan atau interpretasi obyek-obyek tersebut harus mengacu kepada kenampakan di citra Satelit Landsat 8. Karena pengambilan sampel ini harus mengacu kepada visualisasi citra Landsat 8, maka pastikan citra Landsat 8 dimunculkan dalam *map display* dengan men-ceklis layer "TRUE".



Selanjutnya mulailah mengambil polygon sampel dari setiap kelas tutupan lahan. Caranya adalah aktifkan kelas tutupan lahan yang ingin dibuatkan sampelnya. Lalu gunakan tools drawing untuk



Buatlah sampel dengan lebih dari 1 poligon. Semakin lengkap dan representatif sampel yang dibuat, hasil klasifikasi dapat menjadi lebih baik. Maksud representatif disini, artinya sampel yang dibuat terwakili. Sebagai contoh, untuk badan air anda harus mengambil sampel baik di danau, sungai besar maupun laut, serta di laut pun ambilah sampel pada laut dalam dan laut dangkal, hal ini dikarenakan karakteristik nilai pikselnya dapat berbeda walaupun seluruh obyek tersebut masuk dalam kelas yang sama yakni badan air.



Lanjutkan dengan mengambil sampel pada kelas lainnya.

Bila terjadi kesalahan dalam penggambaran sampel, anda dapat menghapus sampel polygon, dengan mengaktifkan tombol pan untuk memilih polygon mana yang akan dihapus. Klik polygon tersebut kemudian akan muncul notifikasi "deleted". Berikut contoh tata caranya.

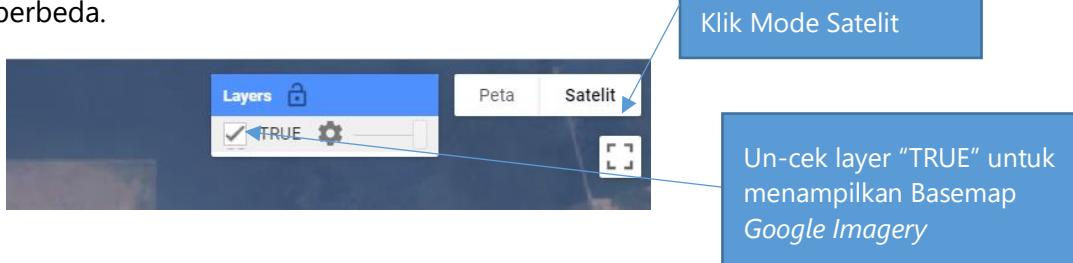




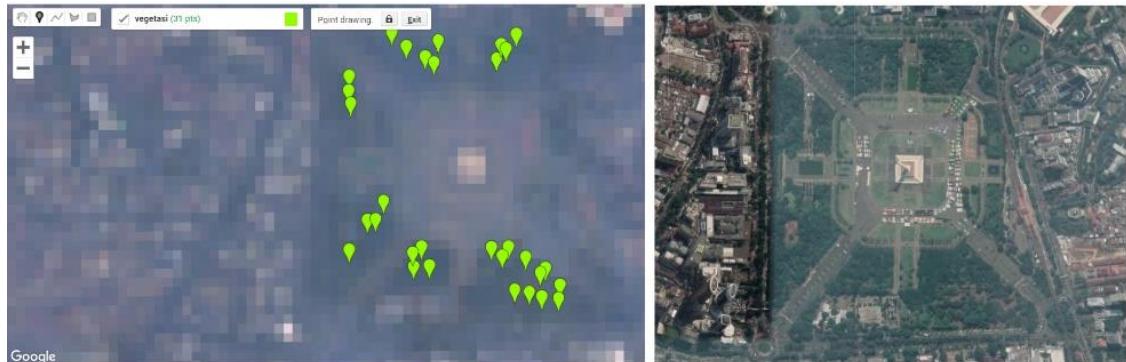
Pengambilan sampel tidak boleh tercampur dengan piksel obyek tutupan lahan lain. Oleh karena itu, penggambaran sampel tidak perlu mengikuti batas kenampakan obyek, cukup piksel yang berada di posisi tengah-tengah obyek tersebut, untuk mengantisipasi tidak tercampur piksel obyek lain.



Kemudian, bila terasa sulit menginterpretasi obyek melalui Citra Landsat 8 karena kurang jelas akibat resolusinya yang rendah, maka anda bisa mengaktifkan mode basemap Satelit. Basemap Satelit ini untuk memunculkan tampilan satelit dari *Google Imagery*, dimana service ini merupakan peta dasar yang terdiri dari kumpulan citra satelit resolusi sangat tinggi. *Google Imagery* ini hanya bersifat alat bantu, namun pengambilan sampel harus mengacu kepada Citra satelit Landsat 8, karena keduanya bisa jadi memiliki waktu perekaman yang berbeda.



Bila obyek terasa sulit untuk diambil, anda dapat memperbesar citra sehingga dapat lebih leluasa dalam memggambar sampel. Namun bila obyek terasa kecil untuk dapat digambar walau sudah diperbesar, anda dapat melakukan penggambaran dengan menggunakan titik .



Landsat 8

Google Imagery

Pastikan semua kelas tutupan lahan sudah memiliki sampel, baik berupa polygon kotak maupun titik.

```

Earth Engine
Search places and datasets...
Get Link Save Run Reset Apps
Projects METRIK
Series ulabsig/latihan1
Imports (8 entries)
var geometry: Polygon, 4 vertices
var vegetasi: FeatureCollection (115 elements)
var badan_air: FeatureCollection (4 elements)
var lahan_terbangun: FeatureCollection (7 elements)
var tambak: FeatureCollection (4 elements)
var rumput_semak: FeatureCollection (5 elements)
var tanah_terbuka: FeatureCollection (7 elements)
var sawah: FeatureCollection (3 elements)

Geometry Imports
geometry (1 poly)
vegetasi (115 pts)
badan_air (4 polys)
lahan_terbangun (7 polys)
tambak (4 polys)
rumput_semak (5 polys)
tanah_terbuka (7 polys)
sawah (3 polys)
+ new layer

```

Selanjutnya kembali ke kolom script, buatlah variabel baru misalkan var **aoi**, yang mendefinsikan gabungan dari seluruh kelas. Masukkan seluruh nama dari kelas tutupan lahan

dengan diikuti fungsi ".merge". Penulisan kelas tutupan lahan haruslah sama seusai dengan penamaan yang terdapat didalam kolom *Geometry imports*.

Kemudian buatlah variabel baru, misalkan var **bands**, yang mendefinsikan band 1 hingga band 7. Sesuai pembahasan pada sub bab sebelumnya, bahwa untuk membedakan antar obyek tutupan lahan diperlukan band-band yang mengandung nilai reflektan saja, yakni terdapat pada band 1 hingga band 7.

Selanjutnya buat kembali variabel baru, misalkan var **training**, untuk mendapatkan karakteristik nilai piksel band 1 hingga band 7 pada setiap sampel kelas tutupan lahan. Masukkan "lc" pada parameter *properties*. Kemudian berikan angka 30 pada parameter *scale* sesuai ukuran piksel Landsat 8.

Berikut adalah script lengkap dari ketiga variabel tersebut.

```
19 var aoi = vegetasi.merge(badan_air).merge(lahan_terbangun).merge(lahan_terbangun)
20 .merge(tambak).merge(rumput_semak).merge(tanah_terbuka).merge(sawah);
21
22 var bands = ['B1','B2','B3','B4','B5','B6','B7'];
23
24 var training = composite.select(bands).sampleRegions({
25   collection: aoi,
26   properties: ['lc'],
27   scale: 30
28 });
29
```

Selanjutnya buatlah variabel baru, misalkan var **classifier** untuk mendefinisikan algoritma yang digunakan. Disini anda dapat memilih beberapa macam algoritma pengklasifikasi, sebagai contoh:

- "cart()" = Classification And Regression Trees
- "svm()" = Suport Vectore Machine
- "randomForest()" = Random Forest, dll.

Selanjutnya buat kembali variabel baru, misalkan var **classified** untuk menjalankan fungsi klasifikasi.

Terakhir tambahkan script "Map.addLayer" untuk memunculkan hasil klasifikasi. Hasil klasifikasi ini akan menghasilkan dataset dengan nilai piksel yang sudah dikelompokkan dalam rentang nilai 0 hingga 6. Nilai ini sesuai kode angka yang diberikan di parameter "lc" saat kita membuat kelas tutupan lahan.

Tabel 6.3. Kelas Tutupan Lahan Hasil Klasifikasi

No	Kelas tutupan lahan	Nilai
1	Vegetasi	0
2	Badan air	1
3	Lahan terbangun	2
4	Tambak	3
5	Rumput/Semak	4
6	Tanah Terbuka	5
7	Sawah	6

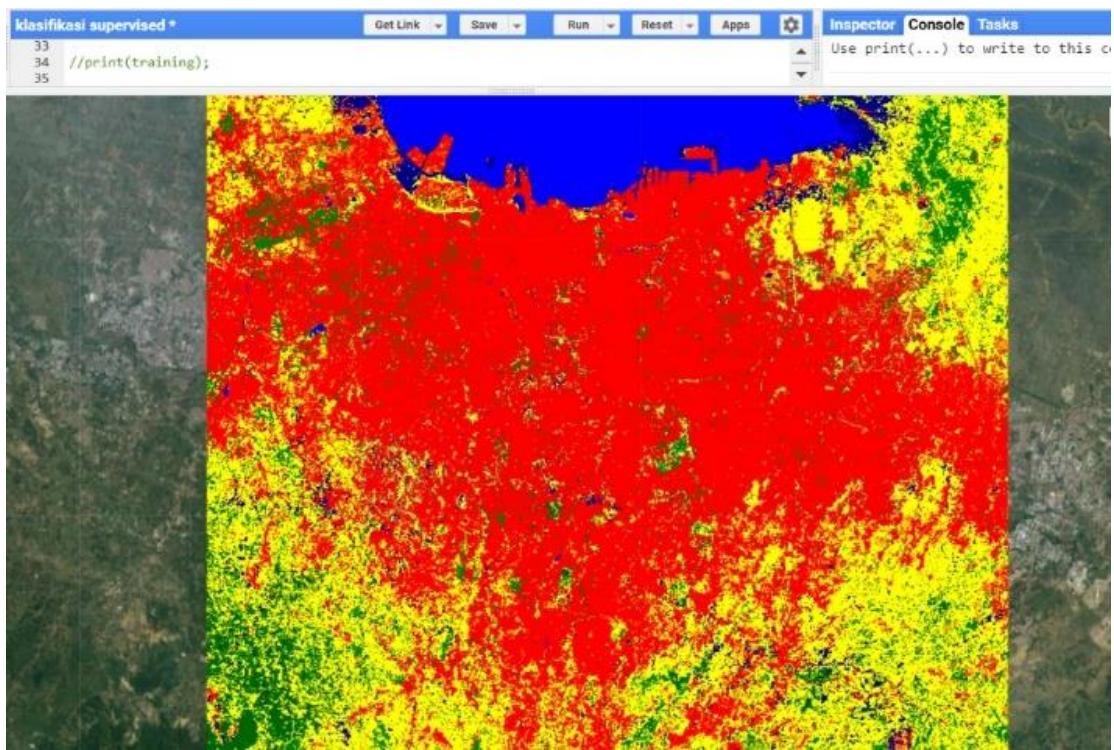
Selanjutnya aturlah warna untuk setiap kelas tutupan lahan. Penamaan warna sebenarnya sudah ada standarisasi dalam bentuk kode, namun anda dapat menggunakan pendekatan bahasa inggris.

Berikut script lengkapnya.

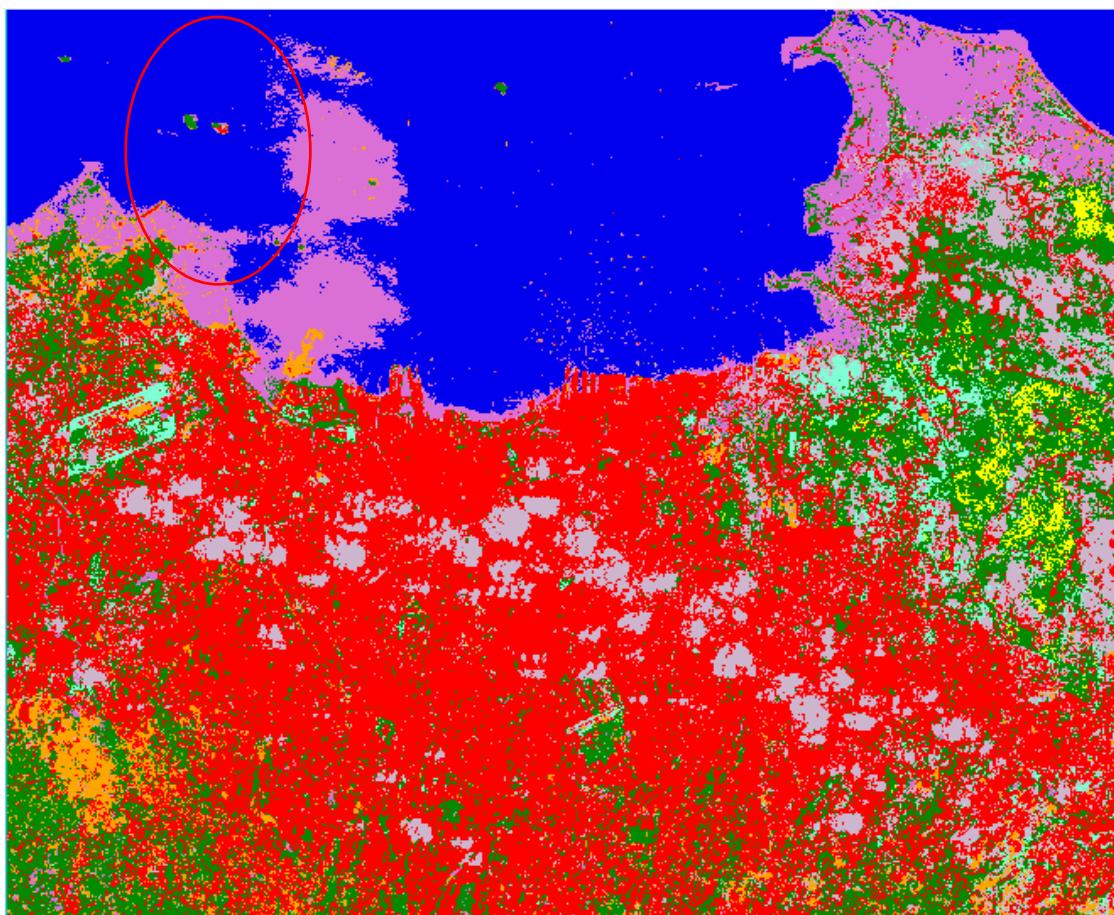
```

Get Link ▾ Save ▾ Run ▾ Reset ▾ Apps
18
19 var aoi = vegetasi.merge(badan_air).merge(lahan_terbangun).merge(lahan_terbangun)
20 .merge(tambak).merge(rumput_semak).merge(tanah_terbuka).merge(sawah);
21
22 var bands = ['B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7'];
23
24 var training = composite.select(bands).sampleRegions({
25   collection: aoi,
26   properties: ['lc'],
27   scale: 30
28 });
29
30 var classifier = ee.Classifier.cart().train({
31   features: training,
32   classProperty: 'lc',
33   inputProperties: bands
34 });
35
36 var classified = composite.select(bands).classify(classifier);
37
38 Map.addLayer(classified,
39 {min: 0, max: 6, palette:['darkgreen','blue','red','darkblue','green','orange','yellow']},
40 'classification');
```

Klik **run** untuk menjalankan proses hingga menghasilkan hasil klasifikasi tutupan lahan seperti gambar di bawah ini.



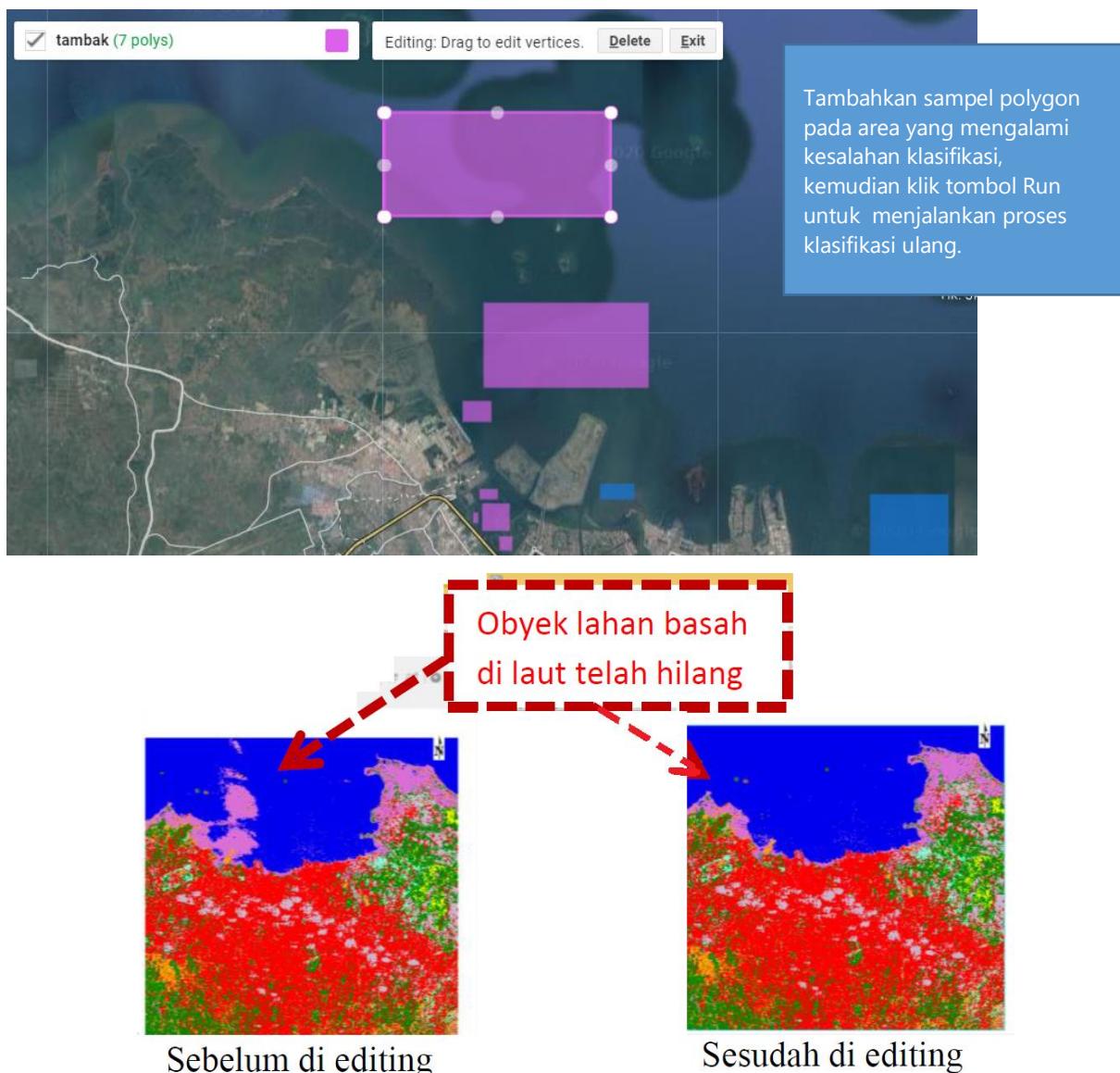
Bila hasil klasifikasi dirasa kurang baik, anda dapat menambahkan sampel polygon.



Sebagai contoh pada gambar diatas ini, area yang dilingkari warna merah, harusnya merupakan laut (biru). Namun dari hasil klasifikasi ini terlihat obyek tersebut adalah lahan

basah (warna ungu). Hal ini terjadi karena tidak ada sampel badan air yang dibuat di area tersebut. Karena ketidak-adaan sampel yang digunakan sebagai acuan, maka piksel-piksel di area tersebut mengambil kriteria sampel yang lebih mendekati, yakni sampel lahan basah. Oleh karena itulah, piksel-piksel tersebut terklasifikasikan menjadi lahan basah.

Untuk dapat memperbaiki hasil klasifikasi ini, tidak perlu anda lakukan dengan membuat ROI dari awal. Anda dapat menambahkan atau mengurangi sampel ROI yang telah anda buat dan kemudian menjalankan kembali proses klasifikasinya dengan mengklik tombol Run.



Selanjutnya anda dapat menyimpan (save) script sehingga anda dapat membukanya kembali dikemudian waktu.

