

# Land Cover Sampling Tool

## Overview

**Land Cover Sampling Tool** dapat digunakan untuk mendesain dan memberikan label pada sampel yang dapat digunakan dalam proses klasifikasi citra penginderaan jauh dengan teknik *supervised machine learning*. Aplikasi ini dibangun pada platform Jupyter Notebook dengan menggunakan beberapa Python package seperti **ee**, **geopandas**, **ipywidgets**, **ipyleaflet**, dan **geemap**. Aplikasi ini terdiri dari dua bagian: 1. membangun set data sampel yang distribusinya didasarkan atas hasil K-Means Clustering dengan proporsi jumlah sesuai luas setiap kluster; 2. memberikan label pada setiap titik sampel secara interaktif melalui *widget ipyleaflet*.

## Background

Peta penutup lahan yang dibuat dengan teknik *supervised machine learning* (misalnya dengan Maximum Likelihood, Support Vector Machine, Classification and Regression Trees, Random Forests, dsb.) akurasi banyak dipengaruhi oleh kualitas sampel yang digunakan. *Labelled samples* dapat diperoleh salah satunya dari hasil observasi lapangan, misalnya plot sampel dan hasil survey. Namun terkadang untuk area pemetaan yang besar, sampel tersebut tidak cukup representatif dari segi kuantitas dan distribusi spasialnya untuk digunakan sebagai sampel dalam model *supervised classification*.

Sampel penutup lahan banyak dibangun dengan membuat data sampel melalui proses observasi dan digitisasi (berupa titik maupun area) pada citra penginderaan jauh resolusi spasial tinggi. Permasalahannya, proses ini seringkali menghasilkan data sampel yang tidak cukup ideal bagi model *supervised classification*. Misalnya, banyak sampel yang *redundant* dan secara statistik tidak meliputi distribusi nilai piksel keseluruhan secara utuh. Artinya kecenderungan tidak terambilnya “kluster” sampel dengan karakteristik tertentu bisa terjadi.

Solusi yang dapat digunakan untuk mengatasi permasalahan tersebut salah satunya adalah pendekatan *semisupervised learning*. Dalam pendekatan *semisupervised learning*, pertama dilakukan proses *unsupervised learning* untuk mengelompokkan piksel menjadi beberapa kluster. Setiap kluster kemudian diberikan label tipe penutup lahan, beberapa di antaranya mungkin masih berupa percampuran beberapa tipe penutup lahan. Melalui observasi, kluster-kluster yang menggambarkan kelas penutup lahan murni kemudian dijadikan acuan untuk membuat *labelled sampel*.

Aplikasi **Land Cover Sampling Tool** mencoba mengadaptasi pendekatan *semisupervised learning* tersebut dengan menggunakan metode K-Means Clustering. Dari setiap kluster yang dihasilkan, sejumlah titik sampel kemudian diambil secara acak dengan jumlah proporsi berdasarkan luas setiap kluster.

Proses berikutnya adalah *data labelling*, yaitu memberikan label pada setiap titik sampel tersebut. Aplikasi ini dapat digunakan untuk berbagai citra multi-spektral maupun hiperspektral, namun dalam prototipe ini digunakan liputan citra Landsat 8 OLI sebagai contoh.

## Prerequisites

Untuk menjalankan **Land Cover Sampling Tool** Jupyter Notebook, pengguna perlu membuat Python virtual environment dan melakukan instalasi beberapa Python package, direkomendasikan menggunakan `conda`.

Pertama, pengguna perlu memastikan bahwa Python dan `anaconda` telah terpasang pada komputer pengguna. Ketika memasang `anaconda`, Python3 otomatis akan terpasang pada komputer pengguna. Instalasi `anaconda` dapat dilihat di sini.

Kedua, pengguna harus memiliki akun Google Earth Engine (sign up).

Pengguna dapat membuat conda environment (lebih detil di sini) dan memasang beberapa Python package dengan command berikut:

```
conda create --name lcms -c conda-forge python=3.8 earthengine-api
geopandas mamba
conda activate lcms
mamba install geemap xarray_leaflet -c conda-forge
```

Pengguna juga dapat memasang Jupyter notebook extension (tidak wajib).

```
conda install jupyter_contrib_nbextensions -c conda-forge
```

## Usage

### Sample generator

**Sample generator** akan membuat *unlabelled samples dataset* yang sebaran dan jumlahnya didasari hasil K-Means Clustering. Pada bagian **User Editable Part** pengguna dapat mengisi variabel jumlah kluster (k) dan jumlah total titik sampel yang ingin dibuat. Titik sampel akan dibuat dengan jumlah proporsional berdasarkan luasan dari setiap kluster. Selain itu pengguna juga perlu mengisi beberapa parameter lain seperti id asset GEE citra yang akan digunakan, area pemetaan, dan folder **Google Drive** untuk ekspor data sampel yang dihasilkan.

```
### USER EDITABLE PART
```

```
# Bagian 1A: nama file object Earth Engine berikut alamat asset-nya
# Contoh: "users/kfaisal/LCMS_Borneo_2016/L8_Borneo_2016_int"
```

```

img_asset = "users/kfaisal/LCMS_Borneo_2016/L8_Borneo_2016_int"

# Bagian 1B: kombinasi RGB
# Bands: {B2 = Blue, B3 = Green, B4 = Red, B5 = NIR, B6 = SWIR1, B7 = SWIR2}
# Pilihan kombinasi RGB yang tersedia: "432" (komposit warna asli), "543", "562",
# "563", "564", "567"
comp_rgb = "562"

# Bagian 2: Study area
# Pilihan: Sumatera, Jawa, Bali_NusaTenggara, Borneo, Sulawesi, Maluku, Papua
studyArea = "Borneo" # jangan lupa membubuhkan tanda petik

# Bagian 3: Jumlah K-Means clusters
num_cluster = 15

# Bagian 4: Jumlah titik sampel yang diinginkan
num_out_samples = 1000

# Bagian 5: Parameter ekspor samples ke Google Drive
folderName = "LCMS_samples"
exportName = "Borneo_2016"

```

Pada tahap terakhir dari bagian ini, *unlabelled samples dataset* yang telah dibuat akan diekspor ke Google Drive Anda pada folder yang telah ditentukan sebelumnya dalam format ESRI Shapefiles. Pengguna perlu mengunduh data tersebut terlebih dahulu untuk dapat melanjutkan ke proses data labelling. Pengguna perlu memastikan untuk menyimpan file unduhan tersebut pada root directory yang sama dengan lokasi file Jupyter notebook disimpan pada komputer pengguna.

## Data labelling

Sebelum memulai proses pemberian label penutup lahan secara interaktif, pengguna perlu mendefinisikan dua variabel pada bagian **User Editable Part** yaitu lokasi penyimpanan titik sampel dalam format ESRI Shapefiles dan pilihan tipe penutup lahan.

```

### USER EDITABLE PART

```

```

# Bagian 1: nama dan folder path lokasi penyimpanan file titik sampel yang dibuat
# berdasarkan hasil clustering dengan K-Means pada tahap sebelumnya
# (ataupun titik yang telah selesai diisi atributnya sebagian)
# Contoh: "./samples/Aug-06-2021_Borneo_2016_stratifiedsamples_15_1000_gcs.shp"
path_to_shp = "./samples/Aug-06-2021_Borneo_2016_stratifiedsamples_15_1000_gcs.shp"

# Bagian 2: daftar label tutupan lahan

```

```
class_opt = ["Forest", "No-forest"]
```

Pada bagian peta interaktif, titik sampel akan ditampilkan satu persatu berdasarkan unique ID yang dimiliki. Pengguna dapat memberikan label dengan memilih salah satu tipe penutup lahan dari *widget dropdownlist* dan meng-klik *trigger button Save change!*. Pengguna dapat beralih ke *feature* berikutnya dengan meng-klik *widget button Next feature* ataupun *Prev. feature*. Pengguna juga dapat beralih ke lokasi *feature* tertentu dengan memilih ID yang diinginkan dan meng-klik *widget button Jump to feature*.

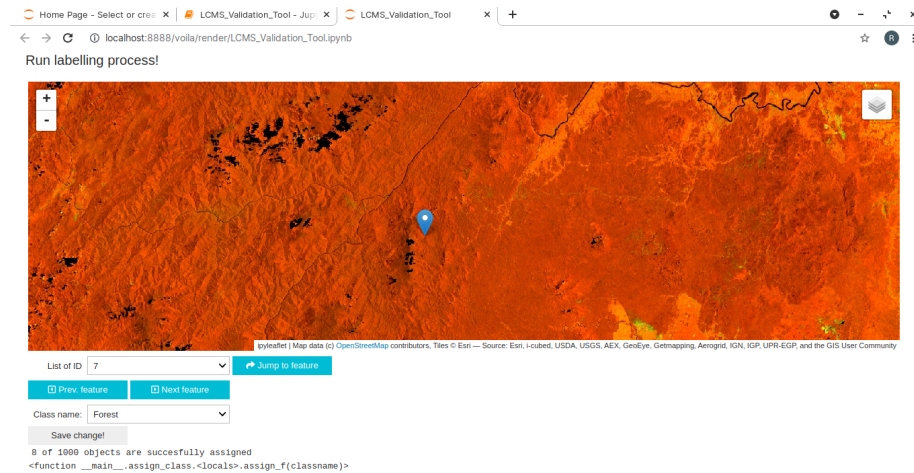


Figure 1: ss\_1

Pengguna dapat menyimpan hasil sementara pemberian label yang dilakukan untuk dapat diedit atau dilanjutkan lagi prosesnya. Pengguna dapat menyimpan file tersebut dengan nama yang sama (*overwrite*) sehingga ketika ingin melanjutkan proses pemberian label pengguna tidak perlu memasukan variabel baru. Pengguna juga dapat mengekspor hasil akhir pemberian label ke GEE asset sehingga data sampel tersebut dapat digunakan untuk proses berikutnya yaitu membangun *supervised machine learning model*.

```
# Export current results
```

```
# Encoding to 0 (No-forest) and 1 (Forest)
```

```
gdf['Class_code'] = numpy.where(gdf['Class'].isnull(), 99, \
numpy.where(gdf['Class'] == 'Forest', 1, 0))
```

```
gdf.to_file(path_to_shp)
```

```
# Export to Asset, later used as machine-learning classification input in GEE
```

```
from datetime import date
```

```
today = date.today()
```

```
todaydate = today.strftime("%b-%d-%Y")
```

```
# Convert geodataframe to ee object
ee_export = geemap.geopandas_to_ee(gdf)

exportTask = ee.batch.Export.table.toAsset(
    collection = ee_export,
    description = str(todaydate) + '_' + studyArea + '_forestCoverSamples',
    assetId = 'users/gemasaktiadzan/' + description)
exportTask.start()
```