/\* ====================================================================================================================

PEMETAAN MANGROVE SENTINEL 2

==================================================================================================================== \*/

Map.centerObject(geometry,12) // Menampilkan Map ke posisi tengah berdasarkan metode AOI (Area Of Interest) dengan perbesaran zoom 12

// Map.setOptions('satellite') // Menampilkan basemap satelit

/\* # Menyiapkan Filter Komposit Citra Sentinel 2 \*/

/\* ## 1.Cloud Masking \*/

//sudah terdapat saluran 'qa' dalam Citra Sentinel 2 yang dapat digunakan untuk membuat fungsi untuk menutupi awan (mask clouds)

function maskClouds(image) {

var qa = image.select('QA60');

// Bit 10 dan 11 masing-masing adalah awan dan cirrus.

var cloudBitMask = 1 << 10;

var cirrusBitMask = 1 << 11;

// Kedua parameter (cloudBitMask dan cirrusBitMask) harus disetel ke nol, menunjukkan kondisi yang bersih.

var mask = qa.bitwiseAnd(cloudBitMask).eq(0)

.and(qa.bitwiseAnd(cirrusBitMask).eq(0));

return image.updateMask(mask).divide(10000);

}

/\* ## 2. Menambahkan Indeks Spektral \*/

// Fungsi ini memetakan indeks spektral yang digunakan untuk Pemetaan Mangrove menggunakan Citra Sentinel 2

var addIndicesS2 = function(img) {

// NDVI

var ndvi = img.normalizedDifference(['B8','B4']).rename('NDVI');

// NDMI (Normalized Difference Mangrove Index - Shi et al 2016 - New spectral metrics for mangrove forest identification)

var ndmi = img.normalizedDifference(['B8','B11']).rename('NDMI');

// MNDWI (Modified Normalized Difference Water Index - Hanqiu Xu, 2006)

var mndwi = img.normalizedDifference(['B11','B3']).rename('MNDWI');

// SR (Simple Ratio)

var sr = img.select('B8').divide(img.select('B4')).rename('SR');

// Band Ratio 54

var ratio54 = img.select('B8').divide(img.select('B4')).rename('R54');

// Band Ratio 35

var ratio35 = img.select('B3').divide(img.select('B8')).rename('R35');

// GCVI

var gcvi = img.expression('(NIR/GREEN)-1',{

'NIR':img.select('B8'),

'GREEN':img.select('B3')

}).rename('GCVI');

return img

.addBands(ndvi)

.addBands(ndmi)

.addBands(mndwi)

.addBands(sr)

.addBands(ratio54)

.addBands(ratio35)

.addBands(gcvi);

};

/\* ## 3. Filter Data Sentinel 2 berdasarkan tanggal \*/

// Pilih tahun perekaman data (ini bisa diubah sesuai dengan tahun yang diinginkan)

var year = 2022;

// Masukkan tanggal awal

var startDate = (year)+'-01-01';

// Masukkan tanggal akhir

var endDate = (year)+'-12-31';

/\* ## 4. Terapkan filter dan masking ke citra Sentinel 2 \*/

var S2 = S2.filterDate(startDate,endDate)

.filter(ee.Filter.lt('CLOUDY\_PIXEL\_PERCENTAGE',10)) //nilai maksimal tutupan awan

.map(maskClouds) // Mask untuk awan dan bayangan awan

.map(addIndicesS2) //Tambahkan indeks

var composite = S2

.median() // penggunaan median reducer

.clip(geometry); // potong composite berdasarkan AOI

// Tambahkan citra komposit 432 atau True Color ke dalam menu tampilan peta

var visualization = {

min: 0.0,

max: 0.3,

bands: ['B4', 'B3', 'B2'],

};

Map.addLayer(composite.clip(geometry), visualization, 'RGB', false)

/\* ## 6. Masking ke area elevasi rendah dan NDVI dan MNDWI \*/

/\* ini tambahan jika diperlukan saja\*/

// var srtmClip = SRTM.clip(geometry); // potong data SRTM sesuai AOI untuk keperluan masking

// var elevationMask = srtmClip.lt(65); // Masking ketinggian kurang dari 65 meter

//Gunakan NDVI dan MNDWI untuk melakukan masking

var NDVIMask = composite.select('NDVI').gt(0.25); //Masking berdasarkan NDVI

var MNDWIMask = composite.select('MNDWI').gt(-0.50); //Masking berdasarkan MNDWI

// Aplikasikan metode masking

var compositeNew = composite

.updateMask(NDVIMask)

.updateMask(MNDWIMask)

// .updateMask(elevationMask)

/\* 7. Tampilkan Hasil \*/

var visPar = {bands:['B8','B12','B3'], min: 0, max: 0.30}; //pilih komposit yang akan ditampilkan

Map.addLayer(compositeNew.clip(geometry), visPar, 'Sentinel 2 Composite', false); //Tambahkan data ke menu tampilan peta

/\* Klasifikasi Menggunakan Random Forest Model \*/

/\* 1. Siapkan training sample data \*/

//buat menggunakan geometry disini sudah disiapkan dengan nama Mangrove dan NonMangrove

// Setelah dilakukan pemilihan training sampel kemudian gabungkan keduanya

var classes = Mangrove.merge(NonMangrove);

//var classes = table; //Gunakan ini jika sudah ada di assets dan sudah disimpan sebelumnya

// Definisikan saluran yang ingin disertakan dalam Random Forest Model

var bands = ['B5','B6','B4','NDVI','MNDWI','SR','GCVI']

// Buatlah variabel dengan nama "image" untuk melakukan pemotongan saluran berdasarkan AOI yang kita tentukan

var image = compositeNew.select(bands).clip(geometry)

// Atur data sampling untuk dimasukkan kedalam model

var samples = image.sampleRegions({

collection: classes, // atur data training sample yang sudah digabungkan sebelumnya

properties: ['landcover'], // label yang digunakan pada data training sample

scale: 10 // ukuran pixel atau resolusi spasial Sentinel

}).randomColumn('random'); // buat kolom dengan nomor acak

// Membagi sample secara acak untuk disisihkan dan digunakan sebagai uji akurasi

var split = 0.8; // DIbagi 80% untuk training sample, 20% untuk uji akurasi

var training = samples.filter(ee.Filter.lt('random', split)); //Subset untuk training sample data

var testing = samples.filter(ee.Filter.gte('random', split)); //Subset untuk uji akurasi

// Print ke dalam console untuk melihat seberapa banyak training sample data yang digunakan

// dan berapa yang digukanan untuk uji akurasi

print('Samples n =', samples.aggregate\_count('.all'));

print('Training n =', training.aggregate\_count('.all'));

print('Testing n =', testing.aggregate\_count('.all'));

/\* Mulai untuk Klasifikasi menggunakan Random Forest Model \*/

//.smileRandomForest digunakan untuk menjalankan model. Di sini dicoba untuk menjalankan model menggunakan 100 pohon

// dan 5 prediktor yang dipilih secara acak per pemisahan ("(100,5)")

var classifier = ee.Classifier.smileRandomForest(100,5).train({

features: training.select(['B5','B6','B4','NDVI','MNDWI','SR','GCVI', 'landcover']), //klasifikasi menggunakan beberapa saluran dan kelas dari training sample data

classProperty: 'landcover', //Menggunakan property : landcover dari training sample data

inputProperties: bands

});

/\* Uji akurasi dari model \*/

var validation = testing.classify(classifier);

var testAccuracy = validation.errorMatrix('landcover', 'classification');

print('Validation error matrix RF: ', testAccuracy);

print('Validation overall accuracy RF: ', testAccuracy.accuracy());

/\* Klasifikasi menggunakan Random Forest Model untuk Citra Sentinel 2\*/

var classifiedrf = image.select(bands) // pilih prediktornya

.classify(classifier); // .classify menerapkan Random Forest Model

//Hasilnya mungkin akan banyak sebaran pixel/noise. Untuk menguranginya buat masking untuk pixel yang tidak terhubung

var pixelcount = classifiedrf.connectedPixelCount(100, false); //Buatlah citra yang menunjukkan jumlah piksel yang terhubung

var countmask = pixelcount.select(0).gt(25); //bersihkan dengan membuang kurang dari 25 pixel yang berhubungan/berdekatan

//Masking hasilnya untuk melihat sebaran mangrove lebih bersih / noiseless

var classMask = classifiedrf.select('classification').gt(0)

var classed= classifiedrf.updateMask(countmask).updateMask(classMask)

/\* Hasil akhir \*/

//Tambahkan hasil klasifikasi ke dalam peta

Map.addLayer (classed, {min: 1, max: 1, palette:'blue'}, 'Mangrove');

/\* Hitung estimasi luasan mangrove \*/

var getArea = classed.multiply(ee.Image.pixelArea()).divide(10000).reduceRegion({

reducer:ee.Reducer.sum(),

geometry:geometry,

scale: 10,

maxPixels:1e13,

tileScale: 16

}).get('classification');

print(getArea, 'Luas Mangrove '+ year + ' dalam Ha') //Tampilkan hasil penghitungan luasan di console

// ekspor hasil klasifikasi ke Google Drive

Export.image.toDrive({

image: classed,

description: 'Mangrove\_'+ year,

region: geometry,

scale: 10,

maxPixels: 1e13

});