

TATA CARA PENGAMBILAN DATA

Data yang berasal dari data citra satelit dikumpulkan melalui tools Google Earth Engine.

- Persiapkan file shp batas administrasi DIY sesuai level (grid/desa/kecamatan/kabkot).
- Import ke script kemudian ganti nama menjadi poi
- Modifikasi script berikut untuk rentang waktu yang sesuai

Rainfall

```
// Load CHIRPS Daily dataset
var dataset = ee.ImageCollection('UCSB-CHG/CHIRPS/DAILY')
    .filterDate('2022-01-01', '2022-12-31')
    .filterBounds(poi)
    .sum()
    .clip(poi);

// Select band 'precipitation' (cuma band ini aja sebenarnya)
var precipitation = dataset.select('precipitation');

//Visualization
var precipitationVis = {
  min: 0,
  max: 4000,
  palette: ['blue', 'lime', 'yellow', 'orange', 'red'],
};

Map.setCenter(110.3688, -7.8026, 8);
Map.addLayer(precipitation, precipitationVis,
'Precipitation');

// Reduksi data citra ke region
var rainfallStats = precipitation.reduceRegions({
  collection: poi,
  reducer: ee.Reducer.mean(),
```

```
    scale: 5000,
  });
// Mengekspor tabel ke Google Drive
Export.table.toDrive({
  collection: rainfallStats,
  description: '2022_Rainfall_DIY',
  folder: 'Yogyakarta',
  fileFormat: 'CSV'
});
```

Land Surface Temperature

```
// Masking fungsi untuk MODIS QA
var masking = function(image) {
  var qa = image.select('QC_Day');
  var Mandatory = 1 << 1;
  var DataQuality = 1 << 3;
  var mask =
qa.bitwiseAnd(Mandatory).eq(0).and(qa.bitwiseAnd(DataQuality).eq(0));
  return image.updateMask(mask);
};

// Filter dataset MODIS untuk tahun 2023 dan masking QA
var dataset = ee.ImageCollection("MODIS/061/MOD11A1")
  .filterDate('2023-01-01', '2023-12-31')
  .filterBounds(poi)
  .map(masking)
  .median()
  .clip(poi);

// Ambil band LST
var landSurfaceTemperature = dataset.select('LST_Day_1km');
// Reduksi ke wilayah administratif (misalnya diy = batas kecamatan)
```

```

var allStats = landSurfaceTemperature.reduceRegions({
  collection: poi,
  reducer: ee.Reducer.median(),
  scale: 500,
}).map(function(feature) {
  var lstKelvin = feature.get('median');
  var lstCelsius = ee.Number(lstKelvin).multiply(0.02).subtract(273.15);
  return feature.set('LST_Celsius', lstCelsius).set('year', 2023);
});

// Ekspor ke Google Drive
Export.table.toDrive({
  collection: allStats,
  description: '2023_LST_DIY',
  folder: 'Yogyakarta',
  fileFormat: 'CSV'
});

```

Sentinel-2: NDVI, NDMI, NDWI, BUI

```

function maskS2clouds(image) {
  var qa = image.select('QA60');

  // Bits 10 and 11 are clouds and cirrus, respectively.
  var cloudBitMask = 1 << 10;
  var cirrusBitMask = 1 << 11;

  // Both flags should be set to zero, indicating clear conditions.
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0)
    .and(qa.bitwiseAnd(cirrusBitMask).eq(0));

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

// 1. Load and preprocess the Sentinel-2 dataset
var dataset = ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED')

```

```

        .filterDate('2023-01-01', '2023-12-31')

        .filterBounds(poi)

        .sort('CLOUD_COVER')

        .map(maskS2clouds)

        .median()

        .clip(poi);

// 2. Select relevant bands
var GREEN = dataset.select('B3');
var RED = dataset.select('B4');
var NIR = dataset.select('B8');
var SWIR = dataset.select('B11');

// 3. Compute indices: NDVI, NDMI, NDWI, and BUI and Clip indices to the
region of interest

var ndvi = NIR.subtract(RED).divide(NIR.add(RED)).rename('NDVI'); //ini
rename untuk mengganti nama band dalam var ndvi
var NDVI = ndvi.clip(poi);

var ndbi = SWIR.subtract(NIR).divide(SWIR.add(NIR)).rename('NDBI');
var NDBI = ndbi.clip(poi);
var bui = NDBI.subtract(NDVI).rename('BUI');
var BUI = bui.clip(poi);
var ndwi = GREEN.subtract(NIR).divide(GREEN.add(NIR)).rename('NDWI');
var NDWI = ndwi.clip(poi);
var ndmi = NIR.subtract(SWIR).divide(NIR.add(SWIR)).rename('NDMI');
var NDMI = ndmi.clip(poi);

// 5. Visualization parameters
var ndviParams = {min: -1, max: 1, palette: ['blue', 'yellow', 'green']};
var buiParams = {min: -1, max: 1, palette: ['blue', 'yellow', 'green']};
var ndwiParams = {min: -1, max: 1, palette: ['blue', 'yellow', 'green']};
var ndmiParams = {min: -1, max: 1, palette: ['blue', 'yellow', 'green']};

// 6. Add layers to the map
Map.setCenter(110.3688, -7.8026, 8);
Map.addLayer(NDVI, ndviParams, 'NDVI');
```

```
Map.addLayer(BUI, buiParams, 'BUI');
Map.addLayer(NDWI, ndwiParams, 'NDWI');
Map.addLayer(NDMI, ndmiParams, 'NDMI');

// 7. Compute statistics for each index in the region
var ndviStats = NDVI.reduceRegions({
  collection: poi,
  reducer: ee.Reducer.median(),
  scale: 10, //ini resolusi spasial sentinel-2 keknya
});
var ndwiStats = NDWI.reduceRegions({
  collection: poi,
  reducer: ee.Reducer.median(),
  scale: 10,
});
var ndmiStats = NDMI.reduceRegions({
  collection: poi,
  reducer: ee.Reducer.median(),
  scale: 10,
});
var buiStats = BUI.reduceRegions({
  collection: poi,
  reducer: ee.Reducer.median(),
  scale: 10,
});

// 8. Export statistics to Google Drive
Export.table.toDrive({
  collection: ndviStats,
  description: '2023_NDVI_DIY',
  folder: 'Ekstraksi Data Tahunan',
  fileFormat: 'CSV'
});
Export.table.toDrive({
```

```

    collection: ndwiStats,
    description: '2023_NDWI_DIY',
    folder: 'Yogyakarta',
    fileFormat: 'CSV'
  });
Export.table.toDrive({
  collection: ndmiStats,
  description: '2023_NDMI_DIY',
  folder: 'Yogyakarta',
  fileFormat: 'CSV'
});
Export.table.toDrive({
  collection: buiStats,
  description: '2023_BUI_DIY',
  folder: 'Yogyakarta',
  fileFormat: 'CSV'
});

```

Sentinel-5P: CO, SO2, NO2

```

Map.setCenter(110.3631, -7.7981, 8);

// Tahun yang ingin dihitung
var year = 2023;

var region = poi; // Ganti 'poi' dengan FeatureCollection wilayahmu
// Filter langsung untuk tahun 2023
var dataset_no2 = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_NO2')
  .filterDate('2023-01-01', '2024-01-01')
  .filterBounds(region)
  .select('NO2_column_number_density')
  .median();

var dataset_so2 = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_SO2')
  .filterDate('2023-01-01', '2024-01-01')
  .filterBounds(region)

```

```

        .select('SO2_column_number_density')
        .median();

var dataset_co = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_CO')
    .filterDate('2023-01-01', '2024-01-01')
    .filterBounds(region)
    .select('CO_column_number_density')
    .median();

// Gabungkan semua band menjadi satu citra
var combinedImage = dataset_no2.addBands(dataset_so2).addBands(dataset_co);

// Hitung nilai median tiap wilayah administratif
var stats = combinedImage.reduceRegions({
    collection: region,
    reducer: ee.Reducer.median(),
    scale: 1000,
}).map(function(feature) {
    return feature.set('year', year);
});

// Ekspor hasilnya ke Google Drive
Export.table.toDrive({
    collection: stats,
    description: '2023_Sentinel5_CO_SO2_NO2_DIY',
    folder: 'Yogyakarta',
    fileFormat: 'CSV'
});

```

NTL

```

Map.setCenter(110.3631, -7.7981, 8);

// Tentukan tahun dan wilayah
var year = 2023;

```

```

var region = poi;
// Filter dataset VIIRS NTL untuk tahun 2023
var datasetNTL = ee.ImageCollection('NOAA/VIIRS/DNB/MONTHLY_V1/VCMSLCFG')
    .filterDate('2023-01-01', '2024-01-01')
    .filterBounds(region)
    .select('avg_rad');

// Hitung median tahunan dari koleksi
var yearlyImage = datasetNTL.median();

// Hitung statistik median NTL per wilayah administratif
var stats = yearlyImage.reduceRegions({
  collection: region,
  reducer: ee.Reducer.median(),
  scale: 500,
}).map(function(feature) {
  return feature.set('year', year);
});

// Ekspor ke Google Drive
Export.table.toDrive({
  collection: stats,
  description: '2023_Yearly_NTL_DIY',
  folder: 'Yogyakarta',
  fileFormat: 'CSV'
});

```

NASA-SRTM: Elevation dan Slope

```

var dataset = ee.Image("USGS/SRTMGL1_003");
var elevation = dataset.select('elevation').clip(poi);
var slope = ee.Terrain.slope(elevation).clip(poi);

Map.addLayer(slope, {min: 0, max: 60}, 'slope');
Map.addLayer(elevation, {min: 0, max: 5000}, 'DEM');

```



```
// Reduksi data citra ke region
var elevationStats = elevation.reduceRegions({
  collection: poi,
  reducer: ee.Reducer.median(),
  scale: 50,
});

var slopeStats = slope.reduceRegions({
  collection: poi,
  reducer: ee.Reducer.median(),
  scale: 50,
});

// Mengekspor tabel ke Google Drive
Export.table.toDrive({
  collection: elevationStats,
  description: '2000_Elevation_DIY',
  folder: 'Yogyakarta',
  fileFormat: 'CSV'
});

Export.table.toDrive({
  collection: slopeStats,
  description: '2000_Slope_DIY',
  folder: 'Yogyakarta',
  fileFormat: 'CSV'
});
```

POI Health Facility Density

POI Density merupakan jumlah fasilitas kesehatan di setiap wilayah, diperoleh dari Sistem Informasi Tuberculosis (SITB).

POI Health Facility Distance

POI Distance merupakan jarak pusat grid ke fasilitas kesehatan terdekat. Untuk level desa, kecamatan, dan kabkot POI Distance dihitung menggunakan rata-rata jarak grid yang berada di dalam area tersebut.

Rasio Fasilitas Kesehatan per Area

Dihitung dari POI Density dibagi dengan luas wilayah. Luas wilayah diperoleh dari data publikasi Badan Pusat Statistik (BPS). Untuk data grid, nilai rasio fasilitas kesehatan per area sama dengan POI Density karena luas permukaan adalah 1 km.

Rasio Fasilitas Kesehatan per 10.000 Penduduk

$$\text{Rasio Faskes per 10.000 penduduk} = \frac{\text{poi density}}{\text{jumlah penduduk}} \times 10.000$$

Jumlah penduduk diperoleh dari publikasi Badan Pusat Statistik (BPS). Untuk data grid, nilai jumlah penduduk didekati dengan data world population yang dihitung melalui:

$$\text{Jumlah Penduduk}_{grid} = \frac{\text{World Pop}_{grid}}{(\text{World Pop}_{kec})} \times \text{Jumlah Penduduk}_{kec}$$

Nilai dari jumlah penduduk world pop telah disediakan dalam template.

PDRB per Kapita

PDRB per kapita diperoleh dari publikasi Badan Pusat Statistik (BPS). Untuk data grid dan kecamatan, nilai PDRB per kapita didekati dengan Relative Wealth Index (RWI) yang dihitung melalui:

$$\text{PDRB}_{per\ kapita_{kec/grid}} = \frac{\text{RWI}_{kec/grid}}{\text{RWI}_{kabkot}} \times \text{PDRB}_{per\ kapita_{kab}}$$

Nilai dari Relative Wealth Index telah disediakan dalam template.

Kepadatan Penduduk

Kepadatan penduduk diperoleh dari publikasi Badan Pusat Statistik (BPS) hingga level desa. Untuk data grid, kepadatan penduduk sama dengan jumlah penduduk dalam satu grid dan didekati dengan data world population yang dihitung melalui:

$$\text{Kepadatan Penduduk}_{grid} = \frac{\text{World Pop}_{grid}}{(\text{World Pop}_{kec})} \times \text{Jumlah Penduduk}_{kec}$$

Nilai dari jumlah penduduk world pop telah disediakan dalam template.