# LAPORAN PRAKTIKUM
# PEMROGRAMAN SPASIAL: WEB
# (SVIG213435)

# ACARA VI
# MEMBANGUN 3D WEBGIS SEDERHANA MENGGUNAKAN CESIUM.JS


**Dibuat oleh:**

| | | |
|---|---|---|
| **Nama** | **:** | **Tengku Feby Mutiansah** |
| **NIM** | **:** | **20/457115/SV/17562** |
| **Hari/Jam** | **:** | **Selasa, 12 April 2022** |
| **Kelompok** | **:** | **PSW** |
| **Asisten** | **:** | **1. Heri Basa Sinaga** |
| | | **2. Aldea Rizka Novareka** |

**PROGRAM STUDI SARJANA TERAPAN**
**SISTEM INFORMASI GEOGRAFIS**
**DEPARTEMEN TEKNOLOGI KEBUMIAN**
**SEKOLAH VOKASI**
**UNIVERSITAS GADJAH MADA**
**YOGYAKARTA**

**2022**

## LEMBAR KERJA PRAKTIKUM

### I. TUJUAN

1. Mengenal teknologi *3D WebGIS*.
2. Membangun *3D WebGIS* menggunakan *Cesium.JS.*

### II. ALAT DAN BAHAN

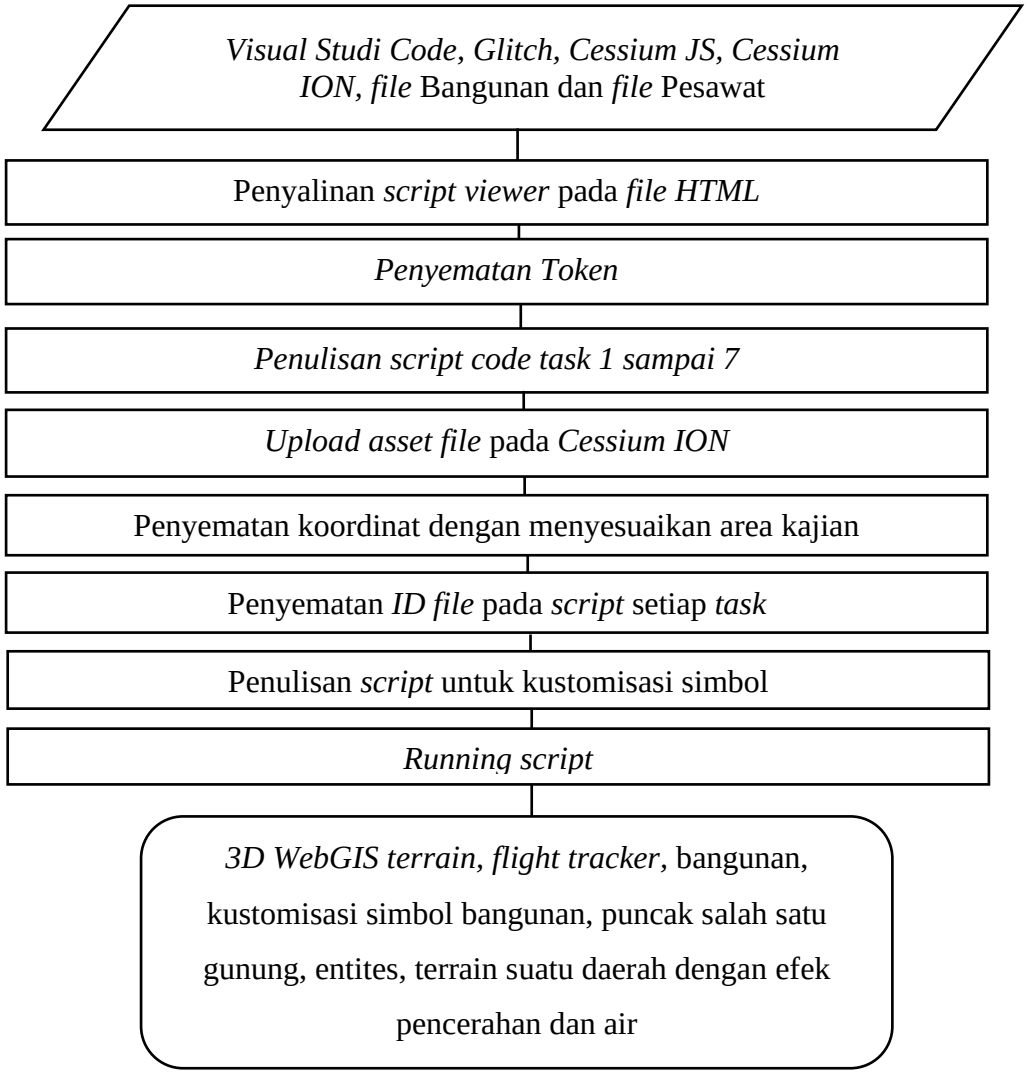
#### II. 1. ALAT

| No | Nama Alat/Bahan | Keterangan |
|---|---|---|
| 1. | Seperangkat Laptop ASUS VivoBook, *64 Bit, Windows 10* | Perangkat keras untuk melakukan praktikum dan membuat laporan praktikum. |
| 2. | *Visual Studi Code* | Perangkat lunak yang digunakan untuk penyusunan *script 3D WebGIS* |
| 3. | *Glitch* | *Platform* yang digunakan untuk penyusunan dan penyimpanan *script 3D WebGIS* |
| 4. | *Cessium JS* | *Library* yang digunakan sebagai acuan dalam membuat *3D WebGIS* |
| 5. | *Cessium ION* | *Server* yang menyimpan data *3D* |
| 6. | Jaringan Internet | Sebagai sarana konektivitas dalam mencari informasi mengenai laporan praktikum. |
| 7. | *Ms. Word* | Perangkat lunak untuk membuat laporan praktikum. |

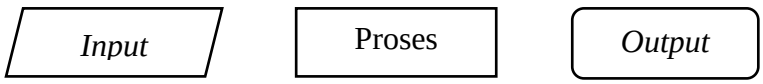
## II. 2. BAHAN

| No. | Nama Bahan | Keterangan |
|---|---|---|
| 1. | *file* Bangunan dan *file* Pesawat Depok | Data yang ditampilkan dalam *3D WebGIS* |
| 2. | *E-book* dan Jurnal | Sebagai referensi dalam memahami materi praktikum |
| 3. | Modul Praktikum | Sebagai acuan dalam melaksanakan praktikum serta dalam menulis laporan praktikum. |

## III. LANGKAH KERJA




Keterangan :

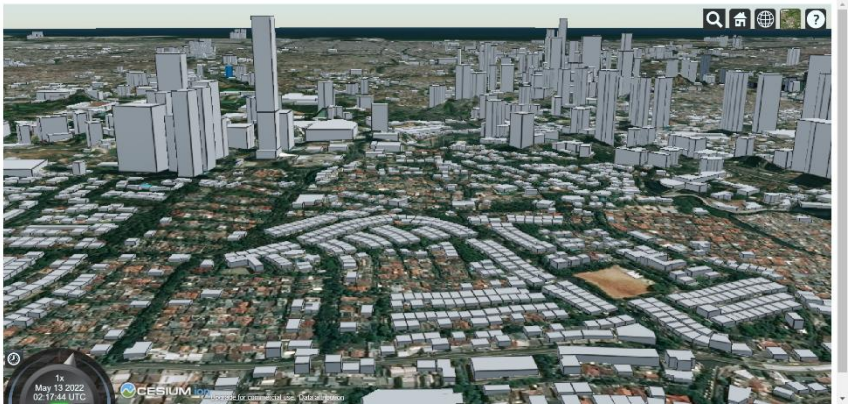


## IV. HASIL PRAKTIKUM DAN PEMBAHASAN

1. Jelaskan teknologi *3D WebGIS* yang diterapkan oleh *Cesium.JS.*

*Cesium* merupakan sebuah *platform end to end* yang dimanfaatkan untuk visualisasi, *tiling*, dan analisis data *3D* geospasial (Wijayanti et al, 2020). Pada dasarnya *Cesium* berfungsi untuk menampilkan data spasial *3D* dalam peta *2D* yang ditampilkan dalam sebuah *web*. *Cesium* memiliki teknologi untuk menujang ditampilkannya *3D* dalam *WebGIS*. Terdapat beberapa platform yang disediakan oleh *Cesium*, diantaranya adalah *Cesium JS* dan *Cesium ION*. *Cesium JS* merupakan sebuah *library* untuk menampilkan *spatial* data *3D* pada peta, agar *WebGIS* memiliki tampilan peta yang interaktif (Cesium GS, 2022). *Cesium ION* merupakan *platform* untuk menyimpan cloud, dan mengoptimalkan konten *WebGIS* sebagai *3D Tiles* (Cesium GS, 2022). Data geospastial *3D* yang ditampilkan pada *Cesium JS* dilakukan dengan cara menyematkan *ID* dari data yang telah di *upload* pada *Cesium ION*. Pembuatan *WebGIS* menggunakan *Cesium* dilakukan dengan mengintegrasikan antara *Cesium JS* dam *Cesium ION*, integrasi tersebut dilakukan dengan menyematkan token dari *Cesium ION* ke dalam baris kode *Cesium JS.*

```
// Initialize the Cesium Viewer in the HTML element with the `cesiumContainer` ID.
const viewer = new Cesium.Viewer('cesiumContainer', {
  terrainProvider: Cesium.createWorldTerrain()
});
// Add Cesium OSM Buildings, a global 3D buildings layer.
const buildingTileset = viewer.scene.primitives.add(Cesium.createOsmBuildings());
// Fly the camera to San Francisco at the given longitude, latitude, and height.
viewer.camera.flyTo({
  destination : Cesium.Cartesian3.fromDegrees(106.80549754400329, -6.2718025077827235, 400),
  orientation : {
    heading : Cesium.Math.toRadians(0.0),
    pitch : Cesium.Math.toRadians(-15.0),
  }
```



Berdasarkan praktikum yang sudah dilakukan, ditampilkan sebuah terrain dengan lokasi sekitar Jakarta Selatan. *Terrain* ini dapat tertampil karena adanya inisialisasi *Cesium Viewer* dengan koordinat sekitar Jakarta Selatan yang disematkan dalam *script. Cesium* juga mampu menampilkan *Flight Tracker*. *Flight tracker* ini pada awalnya hanya ditampilkan rute perjalanan pesawat dengan visualisasi berupa titik-titik *sample,* kemudian dengan menggunakan *Cesium*, titik-titik *sample* tersebut dirubah menjadi model pesawat *3D* yang berjalan menyusuri rute atau sederet titik sample. Teknologi yang disediakan oleh *Cesium* mampu dimanfaatkan untuk menampilkan bangunan pada suatu area dengan customisasi simbol tertentu, bangunan yang ditampilkan pada *WebGIS* juga dapat diatur tampil atau tidaknya dengan menuliskan *script* untuk mengaktifkan *Button Toggle. Cesium* emmiliki teknologi camera untuk melipuut suatu area dengan berotasi meliput area tertentu, contohnya seperti meliput pucak gunung dengan mengitari bagian atas puncak gunung. Dalam *WebGIS* dapat ditampilkan Entites seperti suatu *polygon* yang terletak diatas peta. Teknologi yang terakhir adalah, *Cesium* dapat menampilkan efek pencahayaan dengan inisialisasi menggunakan perintah *requestVertexNormals* dan efek air dengan inisialisasi menggunakan perintah *requestWaterMask.*

2. Kelebihan dan kekurangan *Cesium.JS.*

*Cesium JS* memiliki beberapa kelebihan, diantaranya adalah dapat digunakan untuk menampilkan visualisasi *3D* dari data geospasial. Kemudian *Cesium JS* memiliki fitur-fitur yang dapat menunjang tampilan *WebGIS* seperti fitur *3D tiles*, tampilan *terrain,* tampilan *imagery,* fitur *polygon,* dan *3D models*. Menampilkan geospatial 3D dengan customisasi suatu simbol. Dapat menampilkan *track* dari perjalanan pesawat. Kekurangam *Cesium Js* adalah membutuhkan beberapa *file* tertentu untuk di *download* dan di *upload* pada *asset.* Dalam menentukan posisi 3D dibutuhkan beberapa penyesuaian script agar data 3D yang akan ditampilkan tidak melayang. Dalam menyusun 3D *WebGIS* dibutuhkan *script* yang cukup rumit, sehingga membutuhkan waktu yang lama dalam menyusun *WebGIS* 3D.

## V. KESIMPULAN

1. *WebGIS* dapat ditampilkan dengan visualisasi 3 Dimensi. *3D WebGIS* menampilkan visulisasi data geospasial yang menyerupai objek asli di dunia nyata. Objek 3D yang ditampilkan pada *WebGIS* memiliki ketinggian tertentu. 3D *WebGIS* ditampilkan dengan pengolahan data menggunakan suatu teknologi, salah satunya adalah *Cesium*. Tampilan 3D yang ditampilan pada *WebGIS* dapat berupa sebuah *terrain*, model pesawat, suatu *geometry*, dan sebagainya.
2. *WebGIS* dibuat dengan menggunakan *Cesium. Platform* yang tersedia pada *Cesium* diantaranya adalah *Cesium JS* dan *Cesium ION*. *Cesium JS* merupakan sebuah *library* untuk menampilkan *spatial* data 3D pada peta, agar *WebGIS* memiliki tampilan peta yang interaktif. *Cesium ION* merupakan *platform* untuk menyimpan *cloud,* dan mengoptimalkan konten *WebGIS* sebagai *3D Tiles*. Dibutuhkan integrasi antara *Ceisum JS* dan *Cesium ION* untuk menghasilkan *3d WebGIS,* integrasi tersebut dilakukan dengan menyematkan akses token pada baris kode *Cesium JS*. Pada *WebGIS 3D* dilakukan *upload* data berdasarkan *Cesium ION* yang *ID* dari objeknya di sematkan pula pada *Cesium JS.*

## VI. SARAN

Praktikum *PSWEB* acara VI ini sudah berjalan dengan baik. Terimakasih kepada dosen pengampu praktikum dan asisten praktikum. Belum terdapat saran pada praktikum kali ini, karena praktikum sudah berjalan dengan cukup baik.

## VII. DAFTAR PUSTAKA


Cesium GS. 2022. *3D Geospatial Visualization for the Web*. Online. (https://cesium.com/platform/cesiumjs/, diakses 13 Mei 2022).

Cesium GS. 2022. *Create and Host 3D Content in The Cloud*. Online. (https://cesium.com/platform/cesiumjs/, diakses 13 Mei 2022).

Wijayanti, E. N., Sutanta, H. 2020. *Visualisasi 3D Rencana Detail Tata Ruang Kota Yogyakarta dengan Cesium*. Jurnal Geodesi dan Geomatika. Online. Vol. 3, No. 2, (https://ejournal2.undip.ac.id/index.php/elipsoida/article/view/9212/4973, diakses 13 Mei 2022).

**LAMPIRAN**

Task 1

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <!-- Include the CesiumJS JavaScript and CSS files -->
  <script
src="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Cesium.
js"></script>
  <link
href="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Widge
ts/widgets.css" rel="stylesheet">
</head>
<body>
  <div id="cesiumContainer"></div>
  <script>
    // Your access token can be found at: https://cesium.com/ion/tokens.
    // Replace `your_access_token` with your Cesium ion access token.

    Cesium.Ion.defaultAccessToken =
'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI0NDA1MGExNS00ND
gyLTRkNDUtOTFlNy1hMjJjMjQxOTU5YjYiLCJpZCI6OTA4NzEsImlhdCI6M
TY1MDYxNDA3MH0.FY6Tz0i33Vb7GV0uWZFbZe5-RZOU-
TIc9vsfxv9dtUg';

    // Initialize the Cesium Viewer in the HTML element with the
`cesiumContainer` ID.
    const viewer = new Cesium.Viewer('cesiumContainer', {
      terrainProvider: Cesium.createWorldTerrain()
    });
    // Add Cesium OSM Buildings, a global 3D buildings layer.
    const buildingTileset =
```

```
viewer.scene.primitives.add(Cesium.createOsmBuildings());
    // Fly the camera to San Francisco at the given longitude, latitude, and height.
    viewer.camera.flyTo({
      destination : Cesium.Cartesian3.fromDegrees(106.80549754400329, -
6.2718025077827235, 400),
      orientation : {
        heading : Cesium.Math.toRadians(0.0),
        pitch : Cesium.Math.toRadians(-15.0),
      }
    });
  </script>
 </div>
</body>
</html>
```

Task 2


```
<!DOCTYPE html>
<html lang="en">
<head>
<script
src="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Cesium.
js"></script>
<link
href="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Widge
ts/widgets.css" rel="stylesheet"/>


  <link href="style.css" rel="stylesheet"/>
</head>
<body>
  <div id="cesiumContainer"></div>
```

```
  <script>
    // Get your token from https://cesium.com/ion/tokens
    Cesium.Ion.defaultAccessToken =
'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI0NDA1MGExNS00ND
gyLTRkNDUtOTFlNy1hMjJjMjQxOTU5YjYiLCJpZCI6OTA4NzEsImlhdCI6M
TY1MDYxNDA3MH0.FY6Tz0i33Vb7GV0uWZFbZe5-RZOU-
TIc9vsfxv9dtUg';
    const viewer = new Cesium.Viewer('cesiumContainer');

    const osmBuildings =
viewer.scene.primitives.add(Cesium.createOsmBuildings());

    // STEP 3 CODE (first point)
// This is one of the first radar samples collected for our flight.
const dataPoint = { longitude: -122.38985, latitude: 37.61864, height: -27.32 };
// Mark this location with a red point.
const pointEntity = viewer.entities.add({
  description: `First data point at (${dataPoint.longitude}, ${dataPoint.latitude})`,
  position: Cesium.Cartesian3.fromDegrees(dataPoint.longitude,
dataPoint.latitude, dataPoint.height),
  point: { pixelSize: 10, color: Cesium.Color.RED }
});
// Fly the camera to this point.
viewer.flyTo(pointEntity);

    // STEP 3 CODE (all points)
// These are all the radar points from this flight.
const flightData = JSON.parse(
  '[{"longitude":-122.39053,"latitude":37.61779,"height":-27.32},{"longitude":-
122.39035,"latitude":37.61803,"height":-27.32},{"longitude":-
122.39019,"latitude":37.61826,"height":-27.32},{"longitude":-
122.39006,"latitude":37.6185,"height":-27.32},{"longitude":-
122.38985,"latitude":37.61864,"height":-27.32},{"longitude":-
122.39005,"latitude":37.61874,"height":-27.32},{"longitude":-
122.39027,"latitude":37.61884,"height":-27.32},{"longitude":-
122.39057,"latitude":37.61898,"height":-27.32},{"longitude":-
122.39091,"latitude":37.61912,"height":-27.32},{"longitude":-
122.39121,"latitude":37.61926,"height":-27.32},{"longitude":-
122.39153,"latitude":37.61937,"height":-27.32},{"longitude":-
122.39175,"latitude":37.61948,"height":-27.32},{"longitude":-
122.39207,"latitude":37.6196,"height":-27.32},{"longitude":-
122.39247,"latitude":37.61983,"height":-27.32},{"longitude":-
122.39253,"latitude":37.62005,"height":-27.32},{"longitude":-
122.39226,"latitude":37.62048,"height":-27.32},{"longitude":-
122.39207,"latitude":37.62075,"height":-27.32},{"longitude":-
122.39186,"latitude":37.62106,"height":-27.32},{"longitude":-
```

122.39172,"latitude":37.62127,"height":-27.32},{"longitude":-
122.39141,"latitude":37.62174,"height":-27.32},{"longitude":-
122.39097,"latitude":37.62227,"height":-27.32},{"longitude":-
122.39061,"latitude":37.6224,"height":-27.31},{"longitude":-
122.39027,"latitude":37.62249,"height":-27.31},{"longitude":-
122.38993,"latitude":37.62256,"height":-27.31},{"longitude":-
122.3895,"latitude":37.62267,"height":-27.31},{"longitude":-
122.3889,"latitude":37.62256,"height":-27.31},{"longitude":-
122.38854,"latitude":37.62242,"height":-27.31},{"longitude":-
122.38814,"latitude":37.62225,"height":-27.31},{"longitude":-
122.38778,"latitude":37.62209,"height":-27.31},{"longitude":-
122.38744,"latitude":37.62195,"height":-27.31},{"longitude":-
122.38671,"latitude":37.62164,"height":-27.31},{"longitude":-
122.38638,"latitude":37.62151,"height":-27.31},{"longitude":-
122.38604,"latitude":37.62136,"height":-27.31},{"longitude":-
122.38571,"latitude":37.62123,"height":-27.31},{"longitude":-
122.38512,"latitude":37.62098,"height":-27.31},{"longitude":-
122.3848,"latitude":37.62084,"height":-27.31},{"longitude":-
122.38453,"latitude":37.62074,"height":-27.31},{"longitude":-
122.38425,"latitude":37.62062,"height":-27.3},{"longitude":-
122.38397,"latitude":37.62049,"height":-27.3},{"longitude":-
122.38372,"latitude":37.62039,"height":-27.3},{"longitude":-
122.3835,"latitude":37.6203,"height":-27.3},{"longitude":-
122.38324,"latitude":37.62019,"height":-27.3},{"longitude":-
122.38271,"latitude":37.61996,"height":-27.3},{"longitude":-
122.38248,"latitude":37.61986,"height":-27.3},{"longitude":-
122.38191,"latitude":37.61962,"height":-27.3},{"longitude":-
122.38159,"latitude":37.6195,"height":-27.3},{"longitude":-
122.38126,"latitude":37.61935,"height":-27.3},{"longitude":-
122.38058,"latitude":37.61906,"height":-27.3},{"longitude":-
122.38026,"latitude":37.61894,"height":-27.3},{"longitude":-
122.37955,"latitude":37.61863,"height":-27.3},{"longitude":-
122.37913,"latitude":37.61847,"height":-27.3},{"longitude":-
122.3781,"latitude":37.61813,"height":-27.3},{"longitude":-
122.37763,"latitude":37.61799,"height":-27.29},{"longitude":-
122.37663,"latitude":37.61768,"height":-27.29},{"longitude":-
122.37617,"latitude":37.61753,"height":-27.29},{"longitude":-
122.37561,"latitude":37.61736,"height":-27.29},{"longitude":-
122.37532,"latitude":37.61727,"height":-27.29},{"longitude":-
122.37506,"latitude":37.61718,"height":-27.29},{"longitude":-
122.37468,"latitude":37.61704,"height":-27.29},{"longitude":-
122.37436,"latitude":37.61689,"height":-27.29},{"longitude":-
122.37402,"latitude":37.61674,"height":-27.29},{"longitude":-
122.37362,"latitude":37.61657,"height":-27.29},{"longitude":-
122.3732,"latitude":37.6164,"height":-27.29},{"longitude":-
122.37235,"latitude":37.61604,"height":-27.29},{"longitude":-

122.37198,"latitude":37.61588,"height":-27.29},{"longitude":-
122.3716,"latitude":37.61573,"height":-27.28},{"longitude":-
122.3712,"latitude":37.61555,"height":-27.28},{"longitude":-
122.37032,"latitude":37.61518,"height":-27.28},{"longitude":-
122.36938,"latitude":37.61476,"height":-27.28},{"longitude":-
122.36872,"latitude":37.6143,"height":-27.28},{"longitude":-
122.36811,"latitude":37.61387,"height":-27.28},{"longitude":-
122.36747,"latitude":37.61352,"height":-27.28},{"longitude":-
122.36672,"latitude":37.61321,"height":-27.28},{"longitude":-
122.36602,"latitude":37.61292,"height":-27.28},{"longitude":-
122.3655,"latitude":37.61269,"height":-27.28},{"longitude":-
122.36498,"latitude":37.61247,"height":-27.27},{"longitude":-
122.3639,"latitude":37.61203,"height":-27.27},{"longitude":-
122.36337,"latitude":37.6118,"height":-27.27},{"longitude":-
122.36254,"latitude":37.61146,"height":-27.27},{"longitude":-
122.36203,"latitude":37.61124,"height":-27.27},{"longitude":-
122.36154,"latitude":37.61103,"height":-27.27},{"longitude":-
122.36106,"latitude":37.61082,"height":-27.27},{"longitude":-
122.36029,"latitude":37.61052,"height":-27.27},{"longitude":-
122.35989,"latitude":37.61048,"height":-27.27},{"longitude":-
122.35953,"latitude":37.61055,"height":-27.27},{"longitude":-
122.35922,"latitude":37.61074,"height":-27.26},{"longitude":-
122.35893,"latitude":37.61117,"height":-27.26},{"longitude":-
122.35875,"latitude":37.61144,"height":-27.26},{"longitude":-
122.35865,"latitude":37.61171,"height":-27.26},{"longitude":-
122.35889,"latitude":37.612,"height":-27.26},{"longitude":-
122.35923,"latitude":37.61211,"height":-27.26},{"longitude":-
122.35954,"latitude":37.61222,"height":-27.26},{"longitude":-
122.36029,"latitude":37.61253,"height":-27.27},{"longitude":-
122.36184,"latitude":37.61317,"height":-27.27},{"longitude":-
122.36823,"latitude":37.61588,"height":-27.28},{"longitude":-
122.37163,"latitude":37.61728,"height":-27.28},{"longitude":-
122.38041,"latitude":37.62096,"height":-27.3},{"longitude":-
122.38548,"latitude":37.6231,"height":-27.31},{"longitude":-
122.3905,"latitude":37.6252,"height":10.79},{"longitude":-
122.39517,"latitude":37.62721,"height":56.5},{"longitude":-
122.40037,"latitude":37.62955,"height":125.07},{"longitude":-
122.40507,"latitude":37.63171,"height":186.03},{"longitude":-
122.41457,"latitude":37.63623,"height":292.69},{"longitude":-
122.42428,"latitude":37.64062,"height":384.12},{"longitude":-
122.42896,"latitude":37.64261,"height":429.83},{"longitude":-
122.43398,"latitude":37.6447,"height":483.17},{"longitude":-
122.44357,"latitude":37.64859,"height":582.21},{"longitude":-
122.45319,"latitude":37.65243,"height":643.16},{"longitude":-
122.4639,"latitude":37.65678,"height":719.35},{"longitude":-
122.47395,"latitude":37.66081,"height":803.15},{"longitude":-

122.47922,"latitude":37.66286,"height":841.25},{"longitude":-
122.48441,"latitude":37.66489,"height":879.34},{"longitude":-
122.49505,"latitude":37.66933,"height":970.77},{"longitude":-
122.5003,"latitude":37.67162,"height":993.62},{"longitude":-
122.51083,"latitude":37.67647,"height":1024.08},{"longitude":-
122.51626,"latitude":37.67999,"height":1039.31},{"longitude":-
122.52048,"latitude":37.68431,"height":1046.93},{"longitude":-
122.52323,"latitude":37.68939,"height":1069.79},{"longitude":-
122.52456,"latitude":37.69482,"height":1092.65},{"longitude":-
122.5255,"latitude":37.70298,"height":1115.52},{"longitude":-
122.52602,"latitude":37.70943,"height":1138.38},{"longitude":-
122.52634,"latitude":37.71574,"height":1161.25},{"longitude":-
122.51643,"latitude":37.73744,"height":1245.12},{"longitude":-
122.50723,"latitude":37.74465,"height":1283.25},{"longitude":-
122.50024,"latitude":37.74957,"height":1313.75},{"longitude":-
122.4864,"latitude":37.75959,"height":1397.6},{"longitude":-
122.47231,"latitude":37.77008,"height":1504.31},{"longitude":-
122.45805,"latitude":37.78079,"height":1626.25},{"longitude":-
122.44339,"latitude":37.79173,"height":1755.82},{"longitude":-
122.42878,"latitude":37.80263,"height":1893.01},{"longitude":-
122.41031,"latitude":37.81632,"height":2045.44},{"longitude":-
122.39535,"latitude":37.82744,"height":2175},{"longitude":-
122.3808,"latitude":37.8382,"height":2304.56},{"longitude":-
122.36244,"latitude":37.85177,"height":2472.23},{"longitude":-
122.34698,"latitude":37.86317,"height":2601.79},{"longitude":-
122.33206,"latitude":37.87422,"height":2716.11},{"longitude":-
122.31736,"latitude":37.88505,"height":2830.42},{"longitude":-
122.29758,"latitude":37.89967,"height":2982.85},{"longitude":-
122.28344,"latitude":37.91011,"height":3089.54},{"longitude":-
122.2672,"latitude":37.9221,"height":3142.9},{"longitude":-
122.25107,"latitude":37.93401,"height":3196.25},{"longitude":-
122.235,"latitude":37.94586,"height":3241.98},{"longitude":-
122.21102,"latitude":37.96354,"height":3318.19},{"longitude":-
122.19532,"latitude":37.97507,"height":3386.77},{"longitude":-
122.17095,"latitude":37.99296,"height":3523.94},{"longitude":-
122.15459,"latitude":38.00497,"height":3630.63},{"longitude":-
122.10188,"latitude":38.04364,"height":3927.88},{"longitude":-
122.05068,"latitude":38.08113,"height":4247.98},{"longitude":-
121.99834,"latitude":38.11944,"height":4552.84},{"longitude":-
121.94544,"latitude":38.15803,"height":4842.46},{"longitude":-
121.89173,"latitude":38.19722,"height":5116.83},{"longitude":-
121.83744,"latitude":38.23672,"height":5368.34},{"longitude":-
121.78076,"latitude":38.27786,"height":5619.88},{"longitude":-
121.67054,"latitude":38.35774,"height":6001.09},{"longitude":-
121.56262,"latitude":38.4366,"height":6488.97},{"longitude":-
121.46884,"latitude":38.52786,"height":6984.57},{"longitude":-

121.37305,"latitude":38.62742,"height":7442.27},{"longitude":-
121.27969,"latitude":38.72305,"height":7869.52},{"longitude":-
121.18384,"latitude":38.82103,"height":8114.16},{"longitude":-
121.0849,"latitude":38.92167,"height":8419.92},{"longitude":-
120.98819,"latitude":39.01978,"height":8664.74},{"longitude":-
120.88629,"latitude":39.12269,"height":8917.28},{"longitude":-
120.77951,"latitude":39.23016,"height":9108.83},{"longitude":-
120.67163,"latitude":39.33836,"height":9376.26},{"longitude":-
120.5672,"latitude":39.44264,"height":9612.99},{"longitude":-
120.4577,"latitude":39.55152,"height":9910.45},{"longitude":-
120.35071,"latitude":39.65748,"height":10040.03},{"longitude":-
120.24274,"latitude":39.76404,"height":10040.03},{"longitude":-
120.12367,"latitude":39.86746,"height":10040.02},{"longitude":-
119.9956,"latitude":39.97069,"height":10040.05},{"longitude":-
119.86682,"latitude":40.07272,"height":10040.2},{"longitude":-
119.74329,"latitude":40.17004,"height":10040.38},{"longitude":-
119.61475,"latitude":40.27084,"height":10040.57},{"longitude":-
119.48225,"latitude":40.37425,"height":10040.77},{"longitude":-
119.36212,"latitude":40.46764,"height":10040.95},{"longitude":-
119.23717,"latitude":40.56432,"height":10041.15},{"longitude":-
119.11108,"latitude":40.66152,"height":10041.4},{"longitude":-
118.9834,"latitude":40.75946,"height":10041.66},{"longitude":-
118.85547,"latitude":40.85713,"height":10041.97},{"longitude":-
118.72528,"latitude":40.95625,"height":10042.26},{"longitude":-
118.59631,"latitude":41.05385,"height":10042.52},{"longitude":-
118.46595,"latitude":41.15213,"height":10042.76},{"longitude":-
118.34192,"latitude":41.24524,"height":10042.98},{"longitude":-
118.21114,"latitude":41.34295,"height":10043.29},{"longitude":-
118.20084,"latitude":41.35063,"height":10043.32},{"longitude":-
118.03749,"latitude":41.47218,"height":10043.8},{"longitude":-
118.02363,"latitude":41.48293,"height":10043.85},{"longitude":-
118.00575,"latitude":41.49732,"height":10043.91},{"longitude":-
117.99279,"latitude":41.50822,"height":10043.95},{"longitude":-
117.88224,"latitude":41.62019,"height":10044.46},{"longitude":-
117.77487,"latitude":41.73303,"height":10044.97},{"longitude":-
117.71091,"latitude":41.79902,"height":10045.22},{"longitude":-
117.54896,"latitude":41.96512,"height":10045.77},{"longitude":-
117.44167,"latitude":42.07448,"height":10046.02},{"longitude":-
117.33009,"latitude":42.18782,"height":10046.24},{"longitude":-
117.22356,"latitude":42.29567,"height":10046.47},{"longitude":-
117.11202,"latitude":42.40792,"height":10046.72},{"longitude":-
117.00414,"latitude":42.51599,"height":10047.01},{"longitude":-
116.89089,"latitude":42.62896,"height":10047.23},{"longitude":-
116.78223,"latitude":42.73691,"height":10047.44},{"longitude":-
116.67059,"latitude":42.84728,"height":10047.47},{"longitude":-
116.56448,"latitude":42.95175,"height":10047.46},{"longitude":-

116.45987,"latitude":43.05432,"height":10047.4},{"longitude":-
116.35005,"latitude":43.16137,"height":10047.34},{"longitude":-
116.24506,"latitude":43.26334,"height":10047.36},{"longitude":-
116.13922,"latitude":43.36574,"height":10047.47},{"longitude":-
116.03102,"latitude":43.46997,"height":10047.65},{"longitude":-
115.91975,"latitude":43.58192,"height":10047.9},{"longitude":-
115.81421,"latitude":43.69277,"height":10048.15},{"longitude":-
115.70801,"latitude":43.80258,"height":10048.42},{"longitude":-
115.59547,"latitude":43.91822,"height":10048.71},{"longitude":-
115.48815,"latitude":44.02803,"height":10048.97},{"longitude":-
115.38025,"latitude":44.13795,"height":10049.29},{"longitude":-
115.26595,"latitude":44.25382,"height":10049.57},{"longitude":-
115.15695,"latitude":44.36372,"height":10049.78},{"longitude":-
115.04876,"latitude":44.47243,"height":10049.92},{"longitude":-
114.94008,"latitude":44.5811,"height":10049.96},{"longitude":-
114.83057,"latitude":44.68991,"height":10049.93},{"longitude":-
114.72199,"latitude":44.79744,"height":10049.89},{"longitude":-
114.61474,"latitude":44.90318,"height":10049.86},{"longitude":-
114.50775,"latitude":45.0081,"height":10049.83},{"longitude":-
114.39586,"latitude":45.11742,"height":10049.89},{"longitude":-
114.28724,"latitude":45.22288,"height":10049.95},{"longitude":-
114.1452,"latitude":45.36018,"height":10050.06},{"longitude":-
114.03063,"latitude":45.47031,"height":10050.15},{"longitude":-
113.91023,"latitude":45.56812,"height":10050.28},{"longitude":-
113.77957,"latitude":45.67081,"height":10050.45},{"longitude":-
113.65418,"latitude":45.76769,"height":10050.63},{"longitude":-
113.52166,"latitude":45.86923,"height":10050.8},{"longitude":-
113.39298,"latitude":45.9674,"height":10050.95},{"longitude":-
113.25732,"latitude":46.07053,"height":10051},{"longitude":-
113.12656,"latitude":46.16925,"height":10050.95},{"longitude":-
112.99599,"latitude":46.26736,"height":10050.88},{"longitude":-
112.85692,"latitude":46.37138,"height":10050.79},{"longitude":-
112.72378,"latitude":46.47044,"height":10050.7},{"longitude":-
112.59017,"latitude":46.56935,"height":10050.71},{"longitude":-
112.45451,"latitude":46.66919,"height":10050.75},{"longitude":-
112.40219,"latitude":46.7075,"height":10050.75},{"longitude":-
112.18308,"latitude":46.86731,"height":10050.72},{"longitude":-
112.04336,"latitude":46.9686,"height":10050.62},{"longitude":-
111.91223,"latitude":47.07069,"height":10050.47},{"longitude":-
111.78719,"latitude":47.17534,"height":10050.26},{"longitude":-
111.65659,"latitude":47.28245,"height":10050.09},{"longitude":-
111.52737,"latitude":47.38783,"height":10049.95},{"longitude":-
111.39635,"latitude":47.49399,"height":10049.81},{"longitude":-
111.26383,"latitude":47.60092,"height":10049.73},{"longitude":-
111.13261,"latitude":47.70612,"height":10049.56},{"longitude":-
110.9972,"latitude":47.81413,"height":10049.38},{"longitude":-

110.86276,"latitude":47.92076,"height":10049.15},{"longitude":-
110.72704,"latitude":48.02771,"height":10048.89},{"longitude":-
110.59324,"latitude":48.13269,"height":10048.71},{"longitude":-
110.45213,"latitude":48.24274,"height":10048.51},{"longitude":-
110.29809,"latitude":48.36214,"height":10048.31},{"longitude":-
110.15818,"latitude":48.46991,"height":10048.12},{"longitude":-
110.01434,"latitude":48.58001,"height":10047.89},{"longitude":-
109.87766,"latitude":48.68405,"height":10047.65},{"longitude":-
109.73201,"latitude":48.79427,"height":10047.4},{"longitude":-
109.59253,"latitude":48.89912,"height":10047.26},{"longitude":-
109.4451,"latitude":49.00946,"height":10047.11},{"longitude":-
109.29828,"latitude":49.12085,"height":10047.1},{"longitude":-
109.15861,"latitude":49.22635,"height":10047.09},{"longitude":-
109.00909,"latitude":49.33844,"height":10047.04},{"longitude":-
108.86092,"latitude":49.44878,"height":10046.98},{"longitude":-
108.71825,"latitude":49.55443,"height":10046.88},{"longitude":-
108.56899,"latitude":49.66429,"height":10046.72},{"longitude":-
108.42393,"latitude":49.77044,"height":10046.57},{"longitude":-
108.27145,"latitude":49.88123,"height":10046.36},{"longitude":-
108.1247,"latitude":49.98719,"height":10046.11},{"longitude":-
107.97197,"latitude":50.09683,"height":10045.75},{"longitude":-
107.82331,"latitude":50.20291,"height":10045.33},{"longitude":-
107.68083,"latitude":50.30452,"height":10044.86},{"longitude":-
107.47946,"latitude":50.46744,"height":10044.09},{"longitude":-
107.34006,"latitude":50.57956,"height":10043.61},{"longitude":-
107.20117,"latitude":50.69032,"height":10043.17},{"longitude":-
107.06024,"latitude":50.80199,"height":10042.81},{"longitude":-
106.91856,"latitude":50.91358,"height":10042.53},{"longitude":-
106.7762,"latitude":51.02502,"height":10042.3},{"longitude":-
106.63352,"latitude":51.136,"height":10042.14},{"longitude":-
106.48335,"latitude":51.25195,"height":10042},{"longitude":-
106.33897,"latitude":51.36282,"height":10041.91},{"longitude":-
106.19385,"latitude":51.47346,"height":10041.81},{"longitude":-
106.05192,"latitude":51.58099,"height":10041.72},{"longitude":-
105.89882,"latitude":51.69621,"height":10041.62},{"longitude":-
105.75058,"latitude":51.80705,"height":10041.51},{"longitude":-
105.60936,"latitude":51.91196,"height":10041.39},{"longitude":-
105.45824,"latitude":52.02352,"height":10041.25},{"longitude":-
105.30106,"latitude":52.13869,"height":10041.07},{"longitude":-
105.14606,"latitude":52.25153,"height":10040.87},{"longitude":-
104.99657,"latitude":52.35965,"height":10040.58},{"longitude":-
104.83852,"latitude":52.47313,"height":10040.27},{"longitude":-
104.68841,"latitude":52.58019,"height":10039.9},{"longitude":-
104.5369,"latitude":52.68745,"height":10039.52},{"longitude":-
104.38322,"latitude":52.79568,"height":10039.14},{"longitude":-
104.36918,"latitude":52.80551,"height":10039.1},{"longitude":-

104.14574,"latitude":52.96129,"height":10038.5},{"longitude":-
104.08882,"latitude":53.00075,"height":10038.34},{"longitude":-
103.66236,"latitude":53.29321,"height":10037.06},{"longitude":-
103.50163,"latitude":53.40198,"height":10036.52},{"longitude":-
103.34328,"latitude":53.50841,"height":10035.99},{"longitude":-
103.18163,"latitude":53.61635,"height":10035.45},{"longitude":-
103.01807,"latitude":53.72468,"height":10034.91},{"longitude":-
102.85287,"latitude":53.83333,"height":10034.47},{"longitude":-
102.6947,"latitude":53.93663,"height":10034.08},{"longitude":-
102.67634,"latitude":53.94859,"height":10034.03},{"longitude":-
102.48911,"latitude":54.06987,"height":10033.65},{"longitude":-
102.2399,"latitude":54.22977,"height":10033.22},{"longitude":-
102.07179,"latitude":54.33659,"height":10032.94},{"longitude":-
101.89569,"latitude":54.44769,"height":10032.64},{"longitude":-
101.70779,"latitude":54.56534,"height":10032.33},{"longitude":-
101.63161,"latitude":54.61276,"height":10032.22},{"longitude":-
100.99348,"latitude":55.0037,"height":10031.42},{"longitude":-
100.81579,"latitude":55.11273,"height":10031.19},{"longitude":-
100.63565,"latitude":55.22264,"height":10030.93},{"longitude":-
100.44937,"latitude":55.33539,"height":10030.6},{"longitude":-
100.2748,"latitude":55.44008,"height":10030.29},{"longitude":-
100.0996,"latitude":55.54445,"height":10030},{"longitude":-
99.92199,"latitude":55.64932,"height":10029.73},{"longitude":-
99.74412,"latitude":55.75351,"height":10029.49},{"longitude":-
99.56592,"latitude":55.85728,"height":10029.14},{"longitude":-
99.39177,"latitude":55.95775,"height":10028.76},{"longitude":-
99.21032,"latitude":56.0617,"height":10028.26},{"longitude":-
99.03151,"latitude":56.16321,"height":10027.69},{"longitude":-
98.85481,"latitude":56.26291,"height":10027.12},{"longitude":-
98.75731,"latitude":56.3175,"height":10026.81},{"longitude":-
92.14444,"latitude":59.53107,"height":10623.76},{"longitude":-
91.9352,"latitude":59.61856,"height":10623.89},{"longitude":-
91.7233,"latitude":59.7063,"height":10624.05},{"longitude":-
91.50816,"latitude":59.79458,"height":10624.26},{"longitude":-
91.29959,"latitude":59.87933,"height":10624.49},{"longitude":-
91.08373,"latitude":59.96636,"height":10624.75},{"longitude":-
90.88204,"latitude":60.06123,"height":10624.98},{"longitude":-
90.70332,"latitude":60.17049,"height":10625.19},{"longitude":-
90.51695,"latitude":60.27779,"height":10625.42},{"longitude":-
90.33527,"latitude":60.38063,"height":10625.63},{"longitude":-
90.149,"latitude":60.48518,"height":10625.84},{"longitude":-
90.02708,"latitude":60.5532,"height":10625.97},{"longitude":-
89.74967,"latitude":60.70626,"height":10626.24},{"longitude":-
89.617,"latitude":60.77884,"height":10626.34},{"longitude":-
87.30364,"latitude":61.97255,"height":10628.64},{"longitude":-
59.43304,"latitude":68.13538,"height":10694.36},{"longitude":-

59.06637,"latitude":68.17969,"height":10695.72},{"longitude":-
58.69851,"latitude":68.22324,"height":10696.85},{"longitude":-
58.32642,"latitude":68.26625,"height":10697.56},{"longitude":-
57.95463,"latitude":68.30823,"height":10697.89},{"longitude":-
57.58284,"latitude":68.34924,"height":10698.01},{"longitude":-
57.21012,"latitude":68.38943,"height":10698.12},{"longitude":-
56.83685,"latitude":68.42862,"height":10698.38},{"longitude":-
56.46272,"latitude":68.46693,"height":10698.76},{"longitude":-
56.08337,"latitude":68.50484,"height":10699.18},{"longitude":-
55.70435,"latitude":68.54182,"height":10699.41},{"longitude":-
55.31946,"latitude":68.57824,"height":10699.27},{"longitude":-
54.92968,"latitude":68.61415,"height":10698.75},{"longitude":-
54.5568,"latitude":68.64748,"height":10698.01},{"longitude":-
54.16822,"latitude":68.6814,"height":10697.2},{"longitude":-
53.78302,"latitude":68.71394,"height":10696.55},{"longitude":-
53.39172,"latitude":68.7459,"height":10696.21},{"longitude":-
52.99648,"latitude":68.77739,"height":10696.22},{"longitude":-
52.60954,"latitude":68.80714,"height":10696.5},{"longitude":-
52.21953,"latitude":68.83614,"height":10696.91},{"longitude":-
51.83844,"latitude":68.86361,"height":10697.44},{"longitude":-
51.44309,"latitude":68.89105,"height":10698.12},{"longitude":-
51.04564,"latitude":68.91775,"height":10698.95},{"longitude":-
50.65205,"latitude":68.94307,"height":10699.93},{"longitude":-
50.26114,"latitude":68.96736,"height":10700.99},{"longitude":-
49.862,"latitude":68.9873,"height":10702.1},{"longitude":-
49.45413,"latitude":68.99833,"height":10703.2},{"longitude":-
49.04689,"latitude":69.00851,"height":10704.25},{"longitude":-
48.65419,"latitude":69.01737,"height":10705.2},{"longitude":-
48.29727,"latitude":69.02459,"height":10706.04},{"longitude":-
29.1449,"latitude":67.84658,"height":11341.65},{"longitude":-
28.98855,"latitude":67.82117,"height":11341.68},{"longitude":-
28.52525,"latitude":67.74468,"height":11341.68},{"longitude":-
28.04757,"latitude":67.66386,"height":11341.81},{"longitude":-
27.71401,"latitude":67.60634,"height":11342.03},{"longitude":-
27.37356,"latitude":67.54655,"height":11342.31},{"longitude":-
27.0469,"latitude":67.48827,"height":11342.56},{"longitude":-
26.72125,"latitude":67.42923,"height":11342.67},{"longitude":-
26.39715,"latitude":67.36942,"height":11342.64},{"longitude":-
26.07518,"latitude":67.30913,"height":11342.52},{"longitude":-
25.7552,"latitude":67.24829,"height":11342.39},{"longitude":-
25.49839,"latitude":67.19872,"height":11342.44},{"longitude":-
25.02888,"latitude":67.10655,"height":11342.85},{"longitude":-
24.71243,"latitude":67.04324,"height":11343.37},{"longitude":-
24.39717,"latitude":66.97925,"height":11343.99},{"longitude":-
24.08335,"latitude":66.91452,"height":11344.62},{"longitude":-
23.77346,"latitude":66.84965,"height":11345.14},{"longitude":-

23.46293,"latitude":66.78374,"height":11345.54},{"longitude":-
23.15693,"latitude":66.71787,"height":11345.83},{"longitude":-
22.84894,"latitude":66.65057,"height":11346.07},{"longitude":-
22.54657,"latitude":66.58357,"height":11346.3},{"longitude":-
22.24051,"latitude":66.51475,"height":11346.53},{"longitude":-
21.94095,"latitude":66.44648,"height":11346.77},{"longitude":-
21.64486,"latitude":66.37798,"height":11347},{"longitude":-
21.34827,"latitude":66.30853,"height":11347.2},{"longitude":-
21.05129,"latitude":66.23799,"height":11347.38},{"longitude":-
20.76073,"latitude":66.168,"height":11347.56},{"longitude":-
20.46514,"latitude":66.09585,"height":11347.79},{"longitude":-
20.17269,"latitude":66.02355,"height":11348.04},{"longitude":-
19.8698,"latitude":65.9599,"height":11348.29},{"longitude":-
19.54587,"latitude":65.91103,"height":11348.51},{"longitude":-
19.22667,"latitude":65.86047,"height":11348.73},{"longitude":-
18.90771,"latitude":65.8089,"height":11348.91},{"longitude":-
18.5854,"latitude":65.75574,"height":11349.05},{"longitude":-
18.26923,"latitude":65.70277,"height":11349.11},{"longitude":-
17.94079,"latitude":65.64685,"height":11349.12},{"longitude":-
17.62601,"latitude":65.59234,"height":11349.1},{"longitude":-
17.31125,"latitude":65.53702,"height":11349.04},{"longitude":-
16.99459,"latitude":65.48039,"height":11348.95},{"longitude":-
16.68549,"latitude":65.42427,"height":11348.86},{"longitude":-
16.37524,"latitude":65.36705,"height":11348.81},{"longitude":-
16.07058,"latitude":65.31001,"height":11348.81},{"longitude":-
15.76197,"latitude":65.25128,"height":11348.82},{"longitude":-
15.45353,"latitude":65.19176,"height":11348.76},{"longitude":-
15.1474,"latitude":65.1318,"height":11348.59},{"longitude":-
14.84413,"latitude":65.07147,"height":11348.29},{"longitude":-
14.5448,"latitude":65.01105,"height":11347.88},{"longitude":-
14.23416,"latitude":64.9474,"height":11347.31},{"longitude":-
13.92803,"latitude":64.88374,"height":11346.75},{"longitude":-
13.63037,"latitude":64.82101,"height":11346.29},{"longitude":-
13.33048,"latitude":64.75685,"height":11345.94},{"longitude":-
13.03505,"latitude":64.69281,"height":11345.71},{"longitude":-
12.74288,"latitude":64.62856,"height":11345.56},{"longitude":-
12.4541,"latitude":64.56418,"height":11345.46},{"longitude":-
12.16604,"latitude":64.49899,"height":11345.34},{"longitude":-
11.86913,"latitude":64.43105,"height":11345.14},{"longitude":-
11.61442,"latitude":64.37192,"height":11344.91},{"longitude":-
10.65684,"latitude":64.14345,"height":11343.62},{"longitude":-
10.36856,"latitude":64.07259,"height":11343.14},{"longitude":-
10.08732,"latitude":64.00272,"height":11342.68},{"longitude":-
9.80299,"latitude":63.93689,"height":11342.26},{"longitude":-
9.51532,"latitude":63.87328,"height":11341.9},{"longitude":-
9.2317,"latitude":63.80892,"height":11341.63},{"longitude":-

8.94171,"latitude":63.74226,"height":11341.42},{"longitude":-
8.68986,"latitude":63.68351,"height":11341.25},{"longitude":-
8.68986,"latitude":63.68351,"height":11341.25},{"longitude":-
8.68986,"latitude":63.68351,"height":11341.25},{"longitude":-
7.92336,"latitude":63.50079,"height":11340.42},{"longitude":-
7.64275,"latitude":63.43227,"height":11339.93},{"longitude":-
7.3626,"latitude":63.36295,"height":11339.44},{"longitude":-
7.07893,"latitude":63.29186,"height":11338.96},{"longitude":-
6.79746,"latitude":63.22041,"height":11338.59},{"longitude":-
6.51996,"latitude":63.14909,"height":11338.35},{"longitude":-
6.2396,"latitude":63.07619,"height":11338.19},{"longitude":-
5.96405,"latitude":63.00343,"height":11338.03},{"longitude":-
5.68573,"latitude":62.92918,"height":11337.78},{"longitude":-
5.41433,"latitude":62.8559,"height":11337.48},{"longitude":-
5.14192,"latitude":62.78137,"height":11337.16},{"longitude":-
4.87172,"latitude":62.70662,"height":11336.81},{"longitude":-
4.59666,"latitude":62.62958,"height":11336.46},{"longitude":-
4.32555,"latitude":62.55266,"height":11336.15},{"longitude":-
4.05406,"latitude":62.47478,"height":11335.77},{"longitude":-
3.78426,"latitude":62.39643,"height":11335.2},{"longitude":-
3.51362,"latitude":62.31692,"height":11334.51},{"longitude":-
3.23642,"latitude":62.23443,"height":11333.7},{"longitude":-
2.95807,"latitude":62.15054,"height":11332.93},{"longitude":-
2.68351,"latitude":62.0668,"height":11332.33},{"longitude":-
2.40287,"latitude":61.98033,"height":11331.99},{"longitude":-
2.12929,"latitude":61.89487,"height":11332.02},{"longitude":-
1.85286,"latitude":61.80753,"height":11332.24},{"longitude":-
1.58244,"latitude":61.72105,"height":11332.59},{"longitude":-
1.29796,"latitude":61.629,"height":11332.95},{"longitude":-
0.99897,"latitude":61.53575,"height":11333.24},{"longitude":-
0.72598,"latitude":61.44937,"height":11333.32},{"longitude":-
0.4629,"latitude":61.36505,"height":11333.18},{"longitude":-
0.20344,"latitude":61.28091,"height":11332.86},{"longitude":0.06082,"latitude":
61.19394,"height":11332.32},{"longitude":0.30075,"latitude":61.10381,"height":
11331.7},{"longitude":0.55559,"latitude":61.00829,"height":11331.04},{"longitu
de":0.80816,"latitude":60.91292,"height":11330.49},{"longitude":1.05645,"latitu
de":60.81812,"height":11330.09},{"longitude":1.20163,"latitude":60.76212,"heig
ht":11329.93},{"longitude":1.55545,"latitude":60.62428,"height":11329.72},{"lo
ngitude":1.7891,"latitude":60.53211,"height":11329.61},{"longitude":2.11271,"la
titude":60.40288,"height":11329.37},{"longitude":2.34208,"latitude":60.31018,"h
eight":11329.1},{"longitude":2.58365,"latitude":60.2115,"height":11328.73},{"lo
ngitude":2.82178,"latitude":60.11334,"height":11328.3},{"longitude":3.05901,"la
titude":60.01431,"height":11327.89},{"longitude":3.30066,"latitude":59.91238,"h
eight":11327.52},{"longitude":3.54318,"latitude":59.80911,"height":11327.27},{
"longitude":3.81699,"latitude":59.69128,"height":11327.18},{"longitude":4.0438
2,"latitude":59.59264,"height":11327.21},{"longitude":4.26781,"latitude":59.494

13,"height":11327.28},{"longitude":4.48981,"latitude":59.39563,"height":11327.33},{"longitude":4.71266,"latitude":59.29577,"height":11327.27},{"longitude":4.9339,"latitude":59.19569,"height":11327.12},{"longitude":5.15121,"latitude":59.09647,"height":11326.93},{"longitude":5.369,"latitude":58.99593,"height":11326.76},{"longitude":5.58712,"latitude":58.89445,"height":11326.64},{"longitude":5.80106,"latitude":58.79399,"height":11326.63},{"longitude":6.01209,"latitude":58.69382,"height":11326.61},{"longitude":6.22302,"latitude":58.59279,"height":11326.54},{"longitude":6.43561,"latitude":58.49014,"height":11326.41},{"longitude":6.64318,"latitude":58.38904,"height":11326.06},{"longitude":6.85901,"latitude":58.28279,"height":11325.58},{"longitude":7.06518,"latitude":58.18048,"height":11325.08},{"longitude":7.33904,"latitude":58.04292,"height":11324.34},{"longitude":7.54349,"latitude":57.93929,"height":11323.87},{"longitude":7.75497,"latitude":57.83101,"height":11323.44},{"longitude":7.96483,"latitude":57.72258,"height":11323.05},{"longitude":8.21414,"latitude":57.59252,"height":11322.7},{"longitude":8.41748,"latitude":57.48535,"height":11322.44},{"longitude":8.61745,"latitude":57.3788,"height":11322.24},{"longitude":8.82158,"latitude":57.26926,"height":11322.04},{"longitude":8.97506,"latitude":57.19446,"height":11321.91},{"longitude":9.00507,"latitude":57.18388,"height":11321.88},{"longitude":9.26585,"latitude":57.09398,"height":10613},{"longitude":9.50849,"latitude":57.00964,"height":9820.32},{"longitude":9.78744,"latitude":56.91142,"height":9141.98},{"longitude":10.01519,"latitude":56.83041,"height":8555.11},{"longitude":10.23368,"latitude":56.75202,"height":8044.42},{"longitude":10.44206,"latitude":56.67646,"height":7510.89},{"longitude":10.6392,"latitude":56.6044,"height":6977.33},{"longitude":10.83072,"latitude":56.53383,"height":6359.92},{"longitude":11.00705,"latitude":56.4684,"height":5826.32},{"longitude":11.18348,"latitude":56.4025,"height":5368.92},{"longitude":11.27506,"latitude":56.36806,"height":5224.04},{"longitude":11.35806,"latitude":56.33677,"height":5223.95},{"longitude":11.43522,"latitude":56.30759,"height":5223.86},{"longitude":11.51525,"latitude":56.27714,"height":5200.91},{"longitude":11.6019,"latitude":56.2441,"height":4804.58},{"longitude":11.69271,"latitude":56.20927,"height":4347.29},{"longitude":11.77829,"latitude":56.17645,"height":4034.79},{"longitude":11.86415,"latitude":56.14343,"height":3806.12},{"longitude":11.94858,"latitude":56.11081,"height":3524.1},{"longitude":11.9781,"latitude":56.0994,"height":3394.53},{"longitude":12.00814,"latitude":56.08772,"height":3295.44},{"longitude":12.03852,"latitude":56.07591,"height":3196.35},{"longitude":12.06931,"latitude":56.06392,"height":3142.99},{"longitude":12.09844,"latitude":56.05261,"height":3120.1},{"longitude":12.12674,"latitude":56.04163,"height":3097.22},{"longitude":12.15362,"latitude":56.03114,"height":3074.33},{"longitude":12.18075,"latitude":56.02059,"height":3051.45},{"longitude":12.20744,"latitude":56.01016,"height":3028.56},{"longitude":12.23328,"latitude":56.00001,"height":2998.06},{"longitude":12.25825,"latitude":55.99019,"height":2944.7},{"longitude":12.28366,"latitude":55.98023,"height":2868.48},{"longitude":12.30761,"latitude":55.97082,"height":2746.55},{"longitude":12.33224,"latitude":55.96115,"height":2616.99},{"longitude":12.35754,"latitude":55.95117,"height":2472.19},{"longitude":12.38201,"latitude":55.94156,"height":2335.01},{"longitude":12.41163,"latitude":55.9299,"height":2190.21},{"longitude":12.43567,"latitude":55.92041,"height":2053.03},{"longitude":

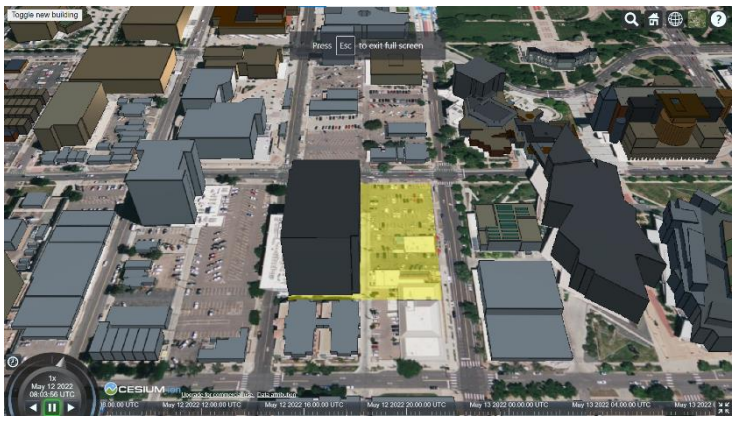
12.46058,"latitude":55.91052,"height":1938.7},{"longitude":12.48321,"latitude":55.90156,"height":1824.38},{"longitude":12.50836,"latitude":55.89155,"height":1694.82},{"longitude":12.53124,"latitude":55.88246,"height":1595.74},{"longitude":12.5542,"latitude":55.87334,"height":1504.28},{"longitude":12.56703,"latitude":55.86827,"height":1458.55},{"longitude":12.58939,"latitude":55.85938,"height":1351.85},{"longitude":12.60165,"latitude":55.85454,"height":1306.12},{"longitude":12.62398,"latitude":55.84569,"height":1214.66},{"longitude":12.63316,"latitude":55.84206,"height":1176.55},{"longitude":12.64551,"latitude":55.83714,"height":1130.81},{"longitude":12.66682,"latitude":55.82863,"height":1062.21},{"longitude":12.68605,"latitude":55.82094,"height":1016.47},{"longitude":12.70785,"latitude":55.81224,"height":955.49},{"longitude":12.72658,"latitude":55.80473,"height":924.99},{"longitude":12.73615,"latitude":55.80093,"height":902.12},{"longitude":12.74526,"latitude":55.79728,"height":871.63},{"longitude":12.76362,"latitude":55.78995,"height":841.13},{"longitude":12.77244,"latitude":55.78638,"height":818.26},{"longitude":12.78259,"latitude":55.78234,"height":795.39},{"longitude":12.80016,"latitude":55.7753,"height":734.41},{"longitude":12.80954,"latitude":55.77102,"height":703.92},{"longitude":12.81495,"latitude":55.7673,"height":673.43},{"longitude":12.82013,"latitude":55.76074,"height":642.94},{"longitude":12.81747,"latitude":55.7384,"height":612.43},{"longitude":12.8148,"latitude":55.73186,"height":597.19},{"longitude":12.81186,"latitude":55.72503,"height":597.18},{"longitude":12.80388,"latitude":55.71316,"height":597.17},{"longitude":12.78018,"latitude":55.69752,"height":581.93},{"longitude":12.77219,"latitude":55.69258,"height":536.21},{"longitude":12.7571,"latitude":55.68304,"height":475.24},{"longitude":12.75097,"latitude":55.67903,"height":429.52},{"longitude":12.727,"latitude":55.66365,"height":307.61},{"longitude":12.6853,"latitude":55.63683,"height":86.64},{"longitude":12.64044,"latitude":55.60784,"height":40.93},{"longitude":12.63927,"latitude":55.60707,"height":40.93},{"longitude":12.63853,"latitude":55.60685,"height":40.93},{"longitude":12.63777,"latitude":55.60692,"height":40.93},{"longitude":12.63715,"latitude":55.60722,"height":40.93},{"longitude":12.63595,"latitude":55.6083,"height":40.94},{"longitude":12.6364,"latitude":55.60869,"height":40.94},{"longitude":12.63698,"latitude":55.60918,"height":40.94},{"longitude":12.63788,"latitude":55.60994,"height":40.94},{"longitude":12.63842,"latitude":55.61042,"height":40.94},{"longitude":12.63904,"latitude":55.61097,"height":40.93},{"longitude":12.64046,"latitude":55.6122,"height":40.93},{"longitude":12.64114,"latitude":55.61282,"height":40.93},{"longitude":12.64187,"latitude":55.61343,"height":40.93},{"longitude":12.64267,"latitude":55.61413,"height":40.93},{"longitude":12.64345,"latitude":55.61482,"height":40.93},{"longitude":12.64487,"latitude":55.61603,"height":40.93},{"longitude":12.64535,"latitude":55.61645,"height":40.93},{"longitude":12.64568,"latitude":55.61689,"height":40.93},{"longitude":12.64554,"latitude":55.61756,"height":40.93},{"longitude":12.64539,"latitude":55.61799,"height":40.93},{"longitude":12.6452,"latitude":55.61862,"height":40.93},{"longitude":12.64503,"latitude":55.61921,"height":40.94},{"longitude":12.64507,"latitude":55.61959,"height":40.94},{"longitude":12.64531,"latitude":55.61997,"height":40.94},{"longitude":12.6457,"latitude":55.62033,"height":40.94},{"longitude":12.64616,"latitude":55.62074,"height":40.94},{"longitude":12.64662,"latitude":55.62112,"height":40.94},{"longitude":12.64732,

```
"latitude":55.62171,"height":40.93},{"longitude":12.64784,"latitude":55.62218,"h
eight":40.93},{"longitude":12.64791,"latitude":55.62269,"height":40.94},{"longit
ude":12.64786,"latitude":55.62305,"height":40.94},{"longitude":12.64782,"latitu
de":55.62333,"height":40.94},{"longitude":12.6478,"latitude":55.62367,"height":
40.94},{"longitude":12.64776,"latitude":55.62396,"height":40.94},{"longitude":1
2.64774,"latitude":55.62442,"height":40.94},{"longitude":12.64774,"latitude":55.
62472,"height":40.94},{"longitude":12.64788,"latitude":55.62494,"height":40.94
},{"longitude":12.6483,"latitude":55.62496,"height":40.94},{"longitude":12.6485
2,"latitude":55.62494,"height":40.94},{"longitude":12.64878,"latitude":55.62493,
"height":40.94},{"longitude":12.64904,"latitude":55.62492,"height":40.94},{"lon
gitude":12.64928,"latitude":55.62491,"height":40.94},{"longitude":12.64936,"lati
tude":55.6249,"height":40.94},{"longitude":12.64949,"latitude":55.62442,"height
":40.94},{"longitude":12.64936,"latitude":55.6247,"height":40.94}]'
);
// Create a point for each.
for (let i = 0; i < flightData.length; i++) {
  const dataPoint = flightData[i];

  viewer.entities.add({
    description: `Location: (${dataPoint.longitude}, ${dataPoint.latitude},
${dataPoint.height})`,
    position: Cesium.Cartesian3.fromDegrees(dataPoint.longitude,
dataPoint.latitude, dataPoint.height),
    point: { pixelSize: 10, color: Cesium.Color.RED }
  });
}

/* Initialize the viewer clock:
  Assume the radar samples are 30 seconds apart, and calculate the entire flight
duration based on that assumption.
  Get the start and stop date times of the flight, where the start is the known flight
departure time (converted from PST
    to UTC) and the stop is the start plus the calculated duration. (Note that Cesium
uses Julian dates. See
    https://simple.wikipedia.org/wiki/Julian_day.)
  Initialize the viewer's clock by setting its start and stop to the flight start and stop
times we just calculated.
  Also, set the viewer's current time to the start time and take the user to that time.
*/
const timeStepInSeconds = 30;
const totalSeconds = timeStepInSeconds * (flightData.length - 1);
const start = Cesium.JulianDate.fromIso8601("2020-03-09T23:10:00Z");
const stop = Cesium.JulianDate.addSeconds(start, totalSeconds, new
Cesium.JulianDate());
viewer.clock.startTime = start.clone();
viewer.clock.stopTime = stop.clone();
```

```
viewer.clock.currentTime = start.clone();
viewer.timeline.zoomTo(start, stop);
// Speed up the playback speed 50x.
viewer.clock.multiplier = 50;
// Start playing the scene.
viewer.clock.shouldAnimate = true;

// The SampledPositionedProperty stores the position and timestamp for each sample along the radar sample series.
const positionProperty = new Cesium.SampledPositionProperty();

for (let i = 0; i < flightData.length; i++) {
  const dataPoint = flightData[i];

  // Declare the time for this individual sample and store it in a new JulianDate instance.
  const time = Cesium.JulianDate.addSeconds(start, i * timeStepInSeconds, new Cesium.JulianDate());
  const position = Cesium.Cartesian3.fromDegrees(dataPoint.longitude, dataPoint.latitude, dataPoint.height);
  // Store the position along with its timestamp.
  // Here we add the positions all upfront, but these can be added at run-time as samples are received from a server.
  positionProperty.addSample(time, position);

  viewer.entities.add({
    description: `Location: (${dataPoint.longitude}, ${dataPoint.latitude}, ${dataPoint.height})`,
    position: position,
    point: { pixelSize: 10, color: Cesium.Color.RED }
  });
}

// STEP 6 CODE (airplane entity)
async function loadModel() {
  // Load the glTF model from Cesium ion.
  const airplaneUri = await Cesium.IonResource.fromAssetId(1013524);
  const airplaneEntity = viewer.entities.add({
    availability: new Cesium.TimeIntervalCollection([ new Cesium.TimeInterval({ start: start, stop: stop }) ]),
    position: positionProperty,
    // Attach the 3D model instead of the green point.
    model: { uri: airplaneUri },
    // Automatically compute the orientation from the position.
    orientation: new Cesium.VelocityOrientationProperty(positionProperty),
    path: new Cesium.PathGraphics({ width: 3 })
```

```
  });

  viewer.trackedEntity = airplaneEntity;
}

loadModel();
  </script>
</body>
</html>
```

Task 3


```
<!DOCTYPE html>
<html lang="en">
<head>
  <script
src="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Cesium.
js"></script>
  <link
href="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Widge
ts/widgets.css" rel="stylesheet">
  <link href="style.css" rel="stylesheet">
```

```
    <style type="text/css">
    #toggle-building { z-index: 1; position: fixed; top: 5px; left: 5px; }
  </style>
</head>
<body>
   <button id="toggle-building">Toggle new building</button>
  <div id="cesiumContainer"></div>
  <script>
    // Your access token can be found at: https://cesium.com/ion/tokens.
    // This is the default access token from your ion account
    Cesium.Ion.defaultAccessToken =
'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI0NDA1MGExNS00ND
gyLTRkNDUtOTFlNy1hMjJjMjQxOTU5YjYiLCJpZCI6OTA4NzEsImlhdCI6M
TY1MDYxNDA3MH0.FY6Tz0i33Vb7GV0uWZFbZe5-RZOU-
TIc9vsfxv9dtUg';

    // Keep your Cesium.Ion.defaultAccessToken = 'your_token_here' line above.
    // STEP 2 CODE
    // Initialize the viewer with Cesium World Terrain.
    const viewer = new Cesium.Viewer('cesiumContainer', {
      terrainProvider: Cesium.createWorldTerrain()
    });
    // Add Cesium OSM Buildings.
    const buildingsTileset =
viewer.scene.primitives.add(Cesium.createOsmBuildings());
    // Fly the camera to Denver, Colorado at the given longitude, latitude, and
height.
    viewer.camera.flyTo({
      destination: Cesium.Cartesian3.fromDegrees(-104.9965, 39.74248, 4000)
    });

    // STEP 3 CODE
async function addBuildingGeoJSON() {
  // Load the GeoJSON file from Cesium ion.
  const geoJSONURL = await Cesium.IonResource.fromAssetId(1013838);
  // Create the geometry from the GeoJSON, and clamp it to the ground.
  const geoJSON = await Cesium.GeoJsonDataSource.load(geoJSONURL, {
clampToGround: true });
  // Add it to the scene.
  const dataSource = await viewer.dataSources.add(geoJSON);
  // By default, polygons in CesiumJS will be draped over all 3D content in the
scene.
  // Modify the polygons so that this draping only applies to the terrain, not 3D
buildings.
  for (const entity of dataSource.entities.values) {
    entity.polygon.classificationType = Cesium.ClassificationType.TERRAIN;
```

```
  }
  // Move the camera so that the polygon is in view.
  viewer.flyTo(dataSource);
}
addBuildingGeoJSON();

    // STEP 4 CODE
// Hide individual buildings in this area using 3D Tiles Styling language.
buildingsTileset.style = new Cesium.Cesium3DTileStyle({
  // Create a style rule to control each building's "show" property.
  show: {
    conditions : [
      // Any building that has this elementId will have `show = false`.
      ['${elementId} === 332469316', false],
      ['${elementId} === 332469317', false],
      ['${elementId} === 235368665', false],
      ['${elementId} === 530288180', false],
      ['${elementId} === 530288179', false],
      // If a building does not have one of these elementIds, set `show = true`.
      [true, true]
    ]
  },
  // Set the default color style for this particular 3D Tileset.
  // For any building that has a `cesium#color` property, use that color, otherwise
make it white.
  color: "Boolean(${feature['cesium#color']}) ? color(${feature['cesium#color']}) :
color('#ffffff')"
});

    // STEP 6 CODE
// Add the 3D Tileset you created from your Cesium ion account.
const newBuildingTileset = viewer.scene.primitives.add(
  new Cesium.Cesium3DTileset({
    url: Cesium.IonResource.fromAssetId(1013991)
  })
);
// Move the camera to the new building.
viewer.flyTo(newBuildingTileset);

    // STEP 7 CODE
// Toggle the tileset's show property when the button is clicked.
document.querySelector('#toggle-building').onclick = function() {
  newBuildingTileset.show = !newBuildingTileset.show;
};
  </script>
</body>
```

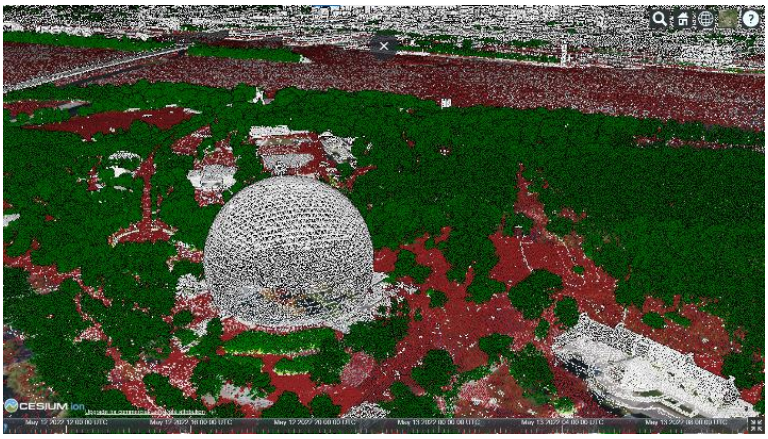
```
</html>
```

Task 4


```
// Grant CesiumJS access to your ion assets
// Cesium.Ion.defaultAccessToken = "_your_cesium_ion_acess_token_";

var viewer = new Cesium.Viewer("cesiumContainer", {
  terrainProvider: Cesium.createWorldTerrain(),
  animation: false,
});

// Adjust the camera to look at Jakpus
viewer.camera.lookAt(
  new Cesium.Cartesian3.fromDegrees(106.8192454001557, -
6.170262304004404),
  new Cesium.Cartesian3(0.0, -1500.0, 1200.0)
);

// Add OSM Building tileset
var osmBuildingsTileset = Cesium.createOsmBuildings();
viewer.scene.primitives.add(osmBuildingsTileset);

// Applying a basic style
function applyBasicStyle() {
  osmBuildingsTileset.style = new Cesium.Cesium3DTileStyle({
    color: {
      conditions: [
        ["${name} === 'Menara BTN'", "color('red')"],
        ["true", "color('white')"],
      ],
    },
```

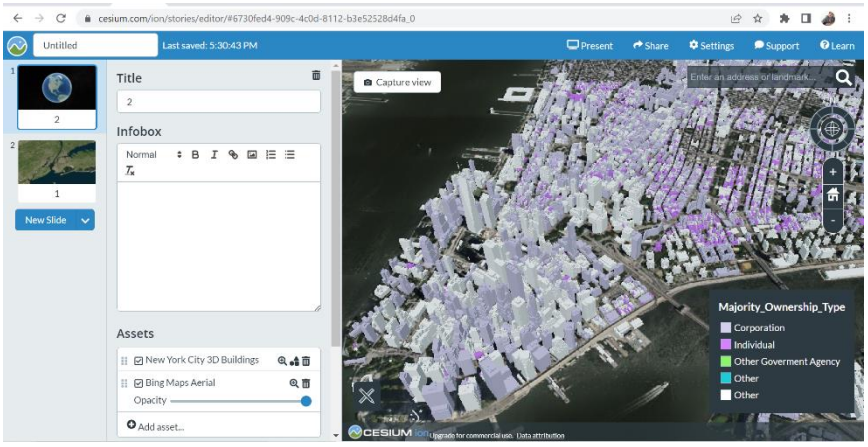
```
  });
}

// Show features based on property
function showFeaturesWithProperty() {
  osmBuildingsTileset.style = new Cesium.Cesium3DTileStyle({
    show: "${feature['building']} === 'residential'||${feature['building']} ===
'appartment'" ,
  });
}

// Color features with conditions
function colorFeaturesWithConditions() {
  osmBuildingsTileset.style = new Cesium.Cesium3DTileStyle({
  defines: {
    distanceFromComplex:
      "distance(vec2(${feature['cesium#longitude']}, ${feature['cesium#latitude']}),
vec2(106.8192454001557, -6.170262304004404))",
  },
  color: {
    conditions: [
      ["${distanceFromComplex} > 0.010", "color('#d65c5c')"],
      ["${distanceFromComplex} > 0.006", "color('#f58971')"],
      ["${distanceFromComplex} > 0.002", "color('#f5af71')"],
      ["${distanceFromComplex} > 0.0001", "color('#f5ec71')"],
      ["true", "color('#ffffff')"],
    ],
  },
});
}

var menu = document.getElementById("dropdown");

menu.options[0].onselect = function () {
  applyBasicStyle();
};

menu.options[1].onselect = function () {
  showFeaturesWithProperty();
};

menu.options[2].onselect = function () {
  colorFeaturesWithConditions();
};

menu.onchange = function () {
```

```
  var item = menu.options[menu.selectedIndex];
  if (item && typeof item.onselect === "function") {
    item.onselect();
  }
};

applyBasicStyle();

menu.selectedIndex = 2;
colorFeaturesWithConditions();
```



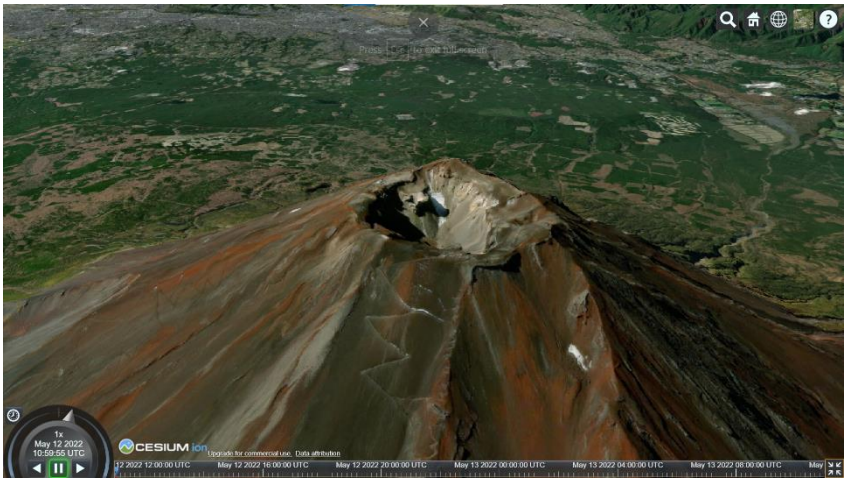
```
var viewer = new Cesium.Viewer("cesiumContainer", {
  terrainProvider: Cesium.createWorldTerrain(),
  animation: false
});

// A ~10 billion point 3D Tileset of the city of Montreal, Canada captured in 2015
with a resolution of 20 cm. Tiled and hosted by Cesium ion.
var pointCloudTileset = viewer.scene.primitives.add(
  new Cesium.Cesium3DTileset({
    url: Cesium.IonResource.fromAssetId(28945),
    pointCloudShading: {
      attenuation: true,
      maximumAttenuation: 2,
    },
  })
);

// Fly to the Biosphere Museum
viewer.camera.setView({
  destination: new Cesium.Cartesian3(
    1269319.8408991008,
```

```
    -4293301.826913256,
    4527724.561372451
   ),
   orientation: {
    direction: new Cesium.Cartesian3(
     -0.742505030107832,
     -0.3413204607149223,
     -0.5763563336703441
    ),
    up: new Cesium.Cartesian3(
     -0.04655102331027917,
     -0.8320643756800384,
     0.5527222421370013
    ),
   },
   easingFunction: Cesium.EasingFunction.QUADRATIC_IN_OUT,
});

pointCloudTileset.style = new Cesium.Cesium3DTileStyle({
  color: {
    conditions: [
      ["${Classification} === 2", "color('brown')"],       // ground
      ["${Classification} === 3", "color('greenyellow')"], // low vegetation
      ["${Classification} === 4", "color('green')"],       // medium vegetation
      ["${Classification} === 5", "color('darkgreen')"],   // high vegetation
      ["true", "color('white')"]
    ]
  }
});
```



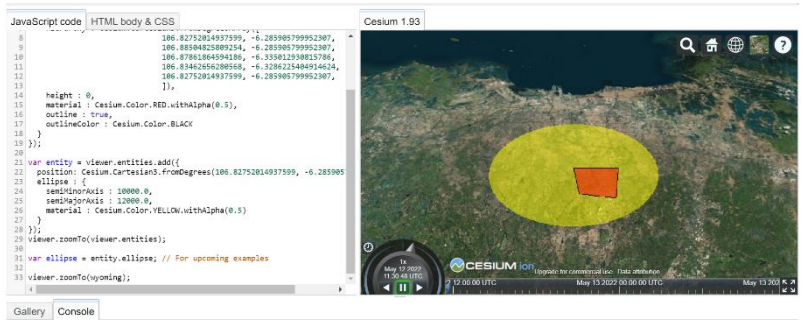

https://cesium.com/ion/stories/viewer/?id=6730fed4-909c-4c0d-8112-b3e52528d4fa

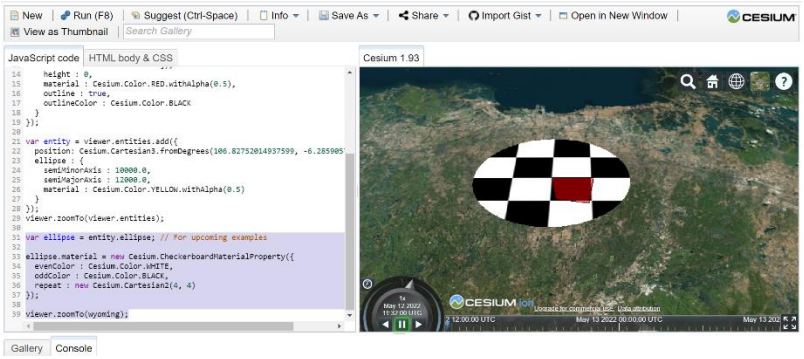
Task 5

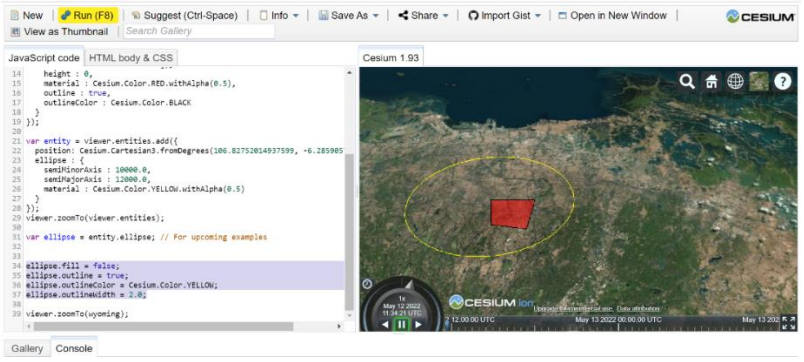


```
// Grant CesiumJS access to your ion assets
// Cesium.Ion.defaultAccessToken = "_your_cesium_ion_acess_token_";

var viewer = new Cesium.Viewer("cesiumContainer", {
  terrainProvider: Cesium.createWorldTerrain(),
});

// Lock camera to a point
var center = Cesium.Cartesian3.fromRadians(2.4213211833389243,
0.6171926869414084, 3626.0426275055174);
var transform = Cesium.Transforms.eastNorthUpToFixedFrame(center);
viewer.scene.camera.lookAtTransform(transform, new
Cesium.HeadingPitchRange(0, -Math.PI/8, 2900));

// Orbit this point
viewer.clock.onTick.addEventListener(function(clock) {
  viewer.scene.camera.rotateRight(0.005);
});

var handler = new Cesium.ScreenSpaceEventHandler(viewer.canvas);
handler.setInputAction(function (event) {
  var pickedPosition = viewer.scene.pickPosition(event.position);
  if (Cesium.defined(pickedPosition)) {
    console.log(pickedPosition);
  }
}, Cesium.ScreenSpaceEventType.LEFT_CLICK);
```

Task 6



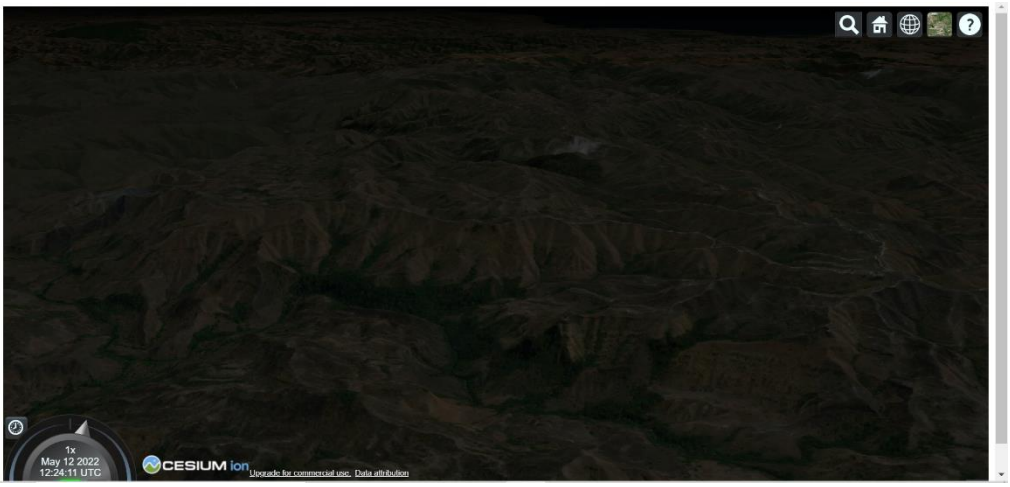




```
const viewer = new Cesium.Viewer("cesiumContainer");
var wyoming = viewer.entities.add({
 polygon : {
  hierarchy : Cesium.Cartesian3.fromDegreesArray([
                    106.82752014937599, -6.285905799952307,
                    106.88504825809254, -6.285905799952307,
                    106.87861864594186, -6.335012930815786,
                    106.83462656280568, -6.3286225404914624,
                    106.82752014937599, -6.285905799952307,
                    ]),
  height : 0,
  material : Cesium.Color.RED.withAlpha(0.5),
  outline : true,
  outlineColor : Cesium.Color.BLACK
 }
```

```
});

var entity = viewer.entities.add({
  position: Cesium.Cartesian3.fromDegrees(106.82752014937599, -
6.285905799952307),
  ellipse : {
    semiMinorAxis : 10000.0,
    semiMajorAxis : 12000.0,
    material : Cesium.Color.YELLOW.withAlpha(0.5)
  }
});
viewer.zoomTo(viewer.entities);

var ellipse = entity.ellipse; // For upcoming examples


ellipse.fill = false;
ellipse.outline = true;
ellipse.outlineColor = Cesium.Color.YELLOW;
ellipse.outlineWidth = 2.0;

viewer.zoomTo(wyoming);
```

Task 7

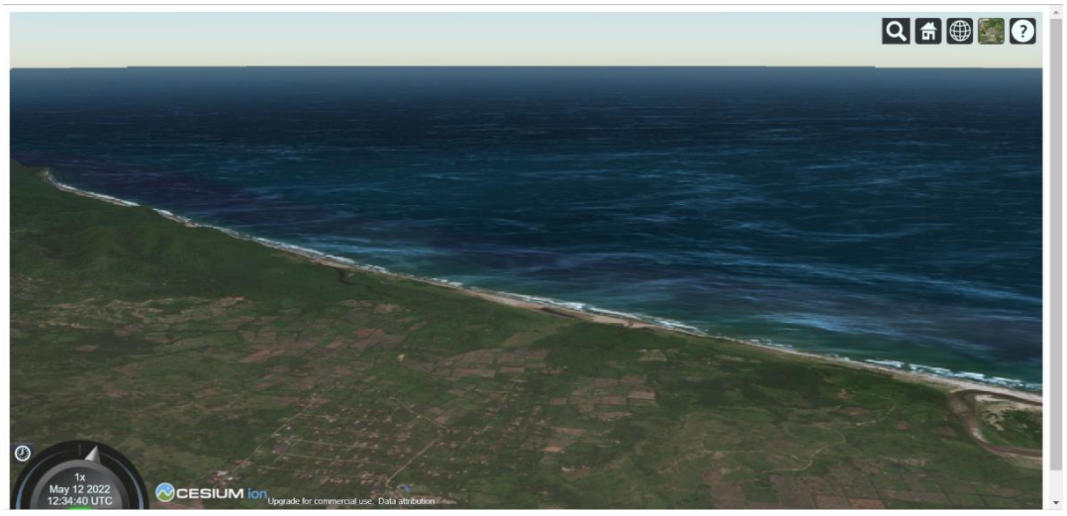
Pencahayaan




```
<!DOCTYPE html>
<html lang="en">

<head>
```

```
  <meta charset="utf-8">
  <!-- Include the CesiumJS JavaScript and CSS files -->
  <script
src="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Cesium.
js"></script>
  <link
href="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Widge
ts/widgets.css" rel="stylesheet">
</head>

<body>
  <div id="cesiumContainer"></div>
  <script>
    // Your access token can be found at: https://cesium.com/ion/tokens.
    // Replace `your_access_token` with your Cesium ion access token.

    Cesium.Ion.defaultAccessToken =
'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI0NDA1MGExNS00ND
gyLTRkNDUtOTFlNy1hMjJjMjQxOTU5YjYiLCJpZCI6OTA4NzEsImlhdCI6M
TY1MDYxNDA3MH0.FY6Tz0i33Vb7GV0uWZFbZe5-RZOU-
TIc9vsfxv9dtUg';

    // Initialize the Cesium Viewer in the HTML element with the
`cesiumContainer` ID.
    var viewer = new Cesium.Viewer('cesiumContainer', {
      terrainProvider: Cesium.createWorldTerrain({
        requestVertexNormals: true
      })
    });
    viewer.scene.globe.enableLighting = true;

    // Fly the camera to Pulau Sumba at the given longitude, latitude, and height.
    viewer.camera.flyTo({
      destination: Cesium.Cartesian3.fromDegrees(120.29449429789291, -
9.903764361255192, 10000),
    orientation : {
        heading : Cesium.Math.toRadians(0.0),
        pitch : Cesium.Math.toRadians(-70.0),
      }
    });
  </script>
  </div>
</body>
</html>
```

Efek air


```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <!-- Include the CesiumJS JavaScript and CSS files -->
  <script
src="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Cesium.
js"></script>
  <link
href="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Widge
ts/widgets.css" rel="stylesheet">
</head>

<body>
  <div id="cesiumContainer"></div>
```

```
  <script>
    // Your access token can be found at: https://cesium.com/ion/tokens.
    // Replace `your_access_token` with your Cesium ion access token.

    Cesium.Ion.defaultAccessToken =
'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI0NDA1MGExNS00NDgyLTRkNDUtOTFlNy1hMjJjMjQxOTU5YjYiLCJpZCI6OTA4NzEsImlhdCI6MTY1MDYxNDA3MH0.FY6Tz0i33Vb7GV0uWZFbZe5-RZOU-TIc9vsfxv9dtUg';

    // Initialize the Cesium Viewer in the HTML element with the
`cesiumContainer` ID.
    var viewer = new Cesium.Viewer('cesiumContainer', {
      terrainProvider: Cesium.createWorldTerrain({
        requestWaterMask: true
      })
    });

    // Fly the camera to Lapaoe Pulau Sumba at the given longitude, latitude, and
height.
    viewer.camera.flyTo({
      destination: Cesium.Cartesian3.fromDegrees(120.02594964869141, -
9.383740861729862, 10000),
      orientation: {
        heading: Cesium.Math.toRadians(0.0),
        pitch: Cesium.Math.toRadians(-70.0),
      }
    });
  </script>
  </div>
</body>
</html>
```

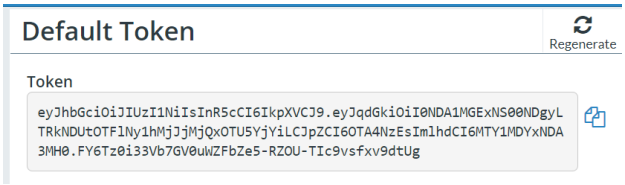
**Langkah Kerja**

**Task 1**

1. Membuat *account* untuk mendapatkan token.

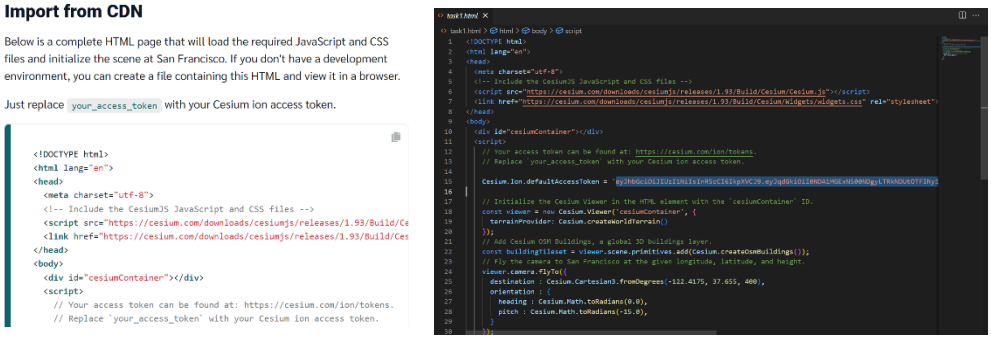

2. Membuka *Acces Tokens tab*.


3. Menyalin token untuk digunakan pada tahap selanjutnya.

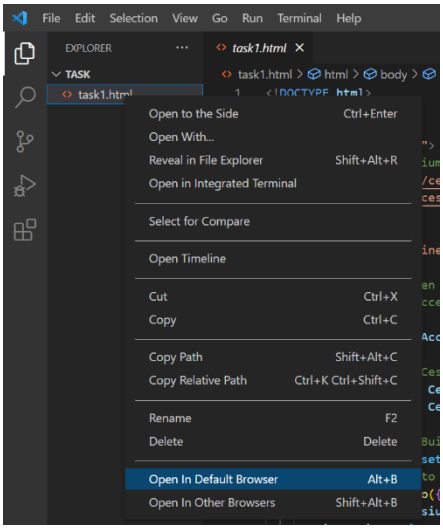



eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI0NDA1MGExNS00NDgyLTRkNDUtOTFlNy1hMjJjMjQxOTU5YjYiLCJpZCI6OTA4NzEsImlhdCI6MTY1MDYxNDA3MH0.FY6Tz0i33Vb7GV0uWZFbZe5-RZOU-TIc9vsfxv9dtUg

4. Menyalin *script* seperti berikut ini ke dalam *visual studio code*, kemudian memasukkan token ke dalam *script*.

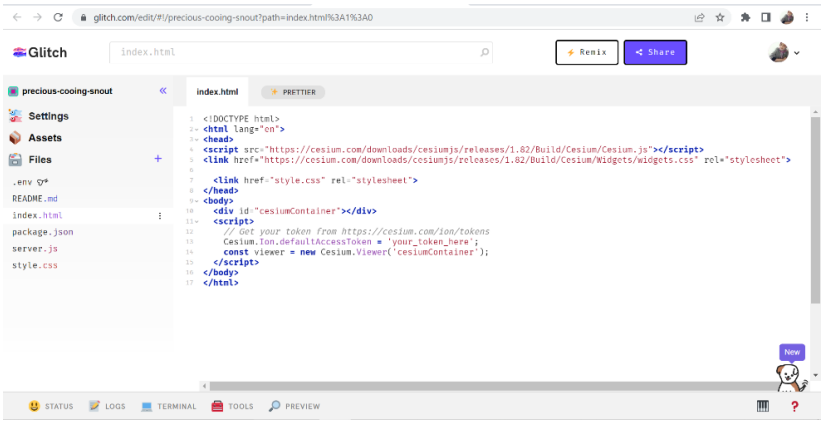



4. Membuka hasilnya pada *web browser*, dengan cara melakukan klik kanan pada *file*, kemudian memilih *open in default folder*, kemudian mengamati hasilnya.

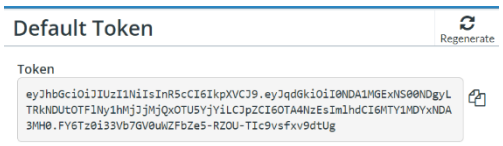



## Task 2

1. Membuat *glitch project*, kemudian membuka *file index.html*.

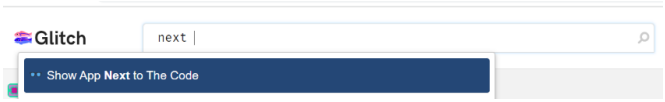


2. Menyalin token, kemudian memasukkan tokennya ke dalam *script.*



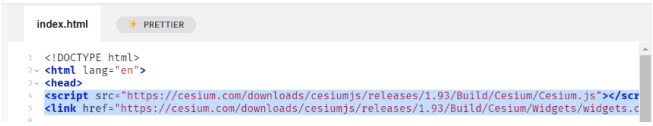

```
<script>
  // Get your token from https://cesium.com/ion/tokens
  Cesium.Ion.defaultAccessToken = 'your_token_here';
  const viewer = new Cesium.Viewer('cesiumContainer');
</script>
```

3. Mencari "*next to the code*" untuk menjalan *script.*

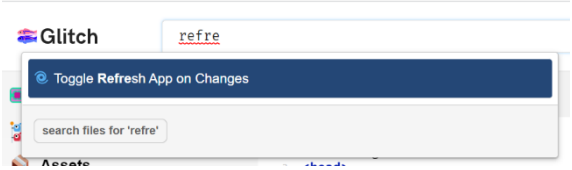



4. Menyalin *script css* dan *js* ke dalam *script glitch.*

```
<script src="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesi
<link href="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesiu
```




5. Mengaktifkan "*refresh app on changes*".




6. Mengetikkan *script* seperti berikut ini untuk menambbahkan *global 3D buildins and terrain.*

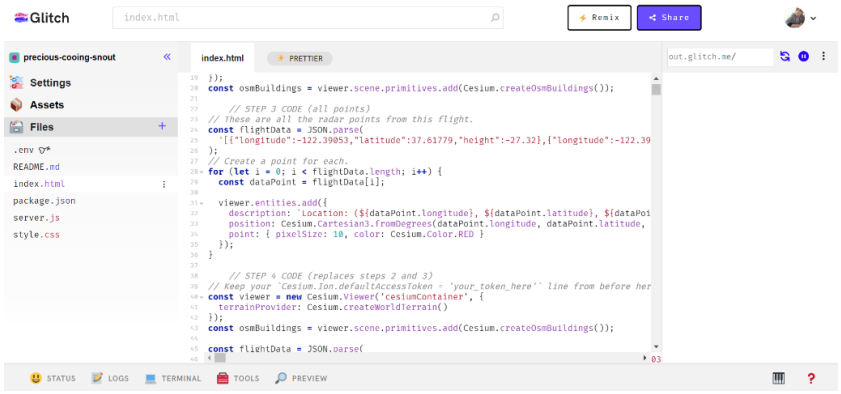
**1. Replace** your JavaScript code in **index.html** with the code below, keeping your access token line from before.

```
// Keep your `Cesium.Ion.defaultAccessToken = 'your_token_here'` line from
const viewer = new Cesium.Viewer('cesiumContainer', {
  terrainProvider: Cesium.createWorldTerrain()
});
const osmBuildings = viewer.scene.primitives.add(Cesium.createOsmBuildings(
```

```
// Keep your `Cesium.Ion.defaultAccessToken = 'your_token_here'` line from before here.
const viewer = new Cesium.Viewer('cesiumContainer', {
  terrainProvider: Cesium.createWorldTerrain()
});
const osmBuildings = viewer.scene.primitives.add(Cesium.createOsmBuildings());
```

7. Menambahkan *script* seperti berikut ini untuk memberikan visualisasi titik pada *scene*.

```
// Create a point for each.
for (let i = 0; i < flightData.length; i++) {
  const dataPoint = flightData[i];

  viewer.entities.add({
    description: `Location: (${dataPoint.longitude}, ${dataPoint.latitude}, ${dataPoint.h
    position: Cesium.Cartesian3.fromDegrees(dataPoint.longitude, dataPoint.latitude, data
    point: { pixelSize: 10, color: Cesium.Color.RED }
  });
}
```

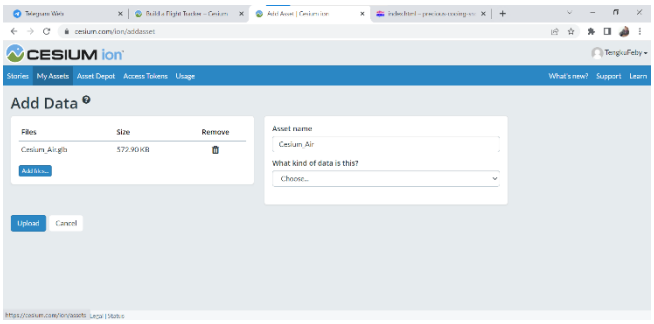




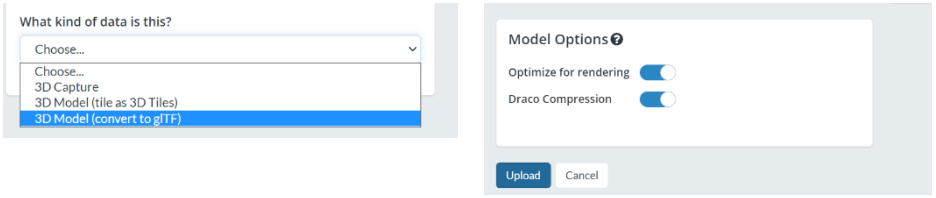

8. Mengunduh *model 3D*.

**1.** Download the 3D model of the airplane.

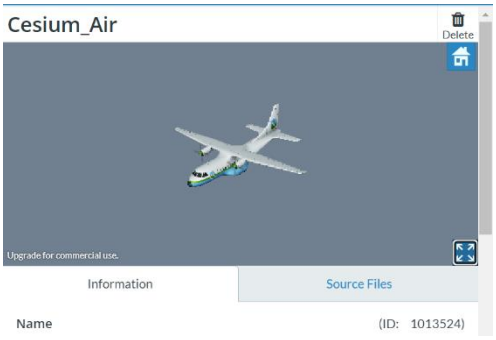
9. Membuka *dashboard cessium*, kemudian melakukan *drag and drop file model 3D* yang telah diunduh.




10. Memilih *3D Model convert togif*, dan melakukan klik pada *upload.*




11. Mencatat *ID* pada data yang telah diupload.




12. Menyalin *script* seperti berikut ini kedalam *file index.html* untuk menggantikan script *STEP 4 CODE (green circle entity).*

```
// STEP 6 CODE (airplane entity)
async function loadModel() {
  // Load the glTF model from Cesium ion.
  const airplaneUri = await Cesium.IonResource.fromAssetId(your_asset_id);
  const airplaneEntity = viewer.entities.add({
    availability: new Cesium.TimeIntervalCollection([ new Cesium.TimeInterval({ start
    position: positionProperty,
    // Attach the 3D model instead of the green point.
    model: { uri: airplaneUri },
    // Automatically compute the orientation from the position.
    orientation: new Cesium.VelocityOrientationProperty(positionProperty),
    path: new Cesium.PathGraphics({ width: 3 })
  });

  viewer.trackedEntity = airplaneEntity;
}

loadModel();
```

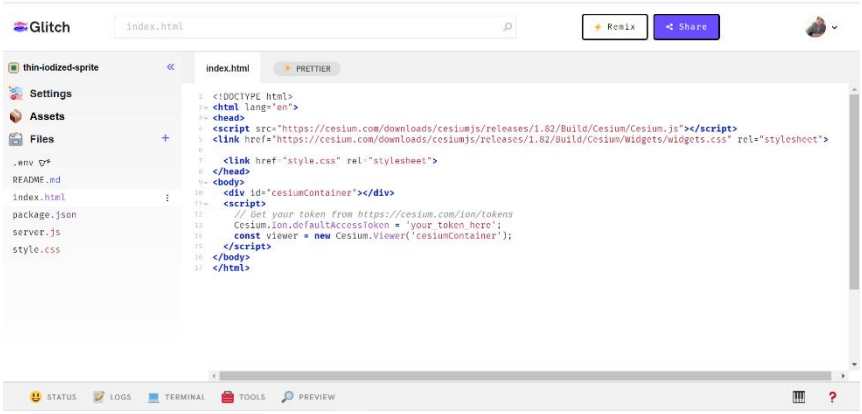
13. Menyalin *ID* dari *dashboard* ke dalam *script.*

```
// STEP 6 CODE (airplane entity)
async function loadModel() {
  // Load the glTF model from Cesium ion.
  const airplaneUri = await Cesium.IonResource.fromAssetId(your_asset_id);
  const airplaneEntity = viewer.entities.add({
    availability: new Cesium.TimeIntervalCollection([ new Cesium.TimeInterva

// STEP 6 CODE (airplane entity)
async function loadModel() {
  // Load the glTF model from Cesium ion.
  const airplaneUri = await Cesium.IonResource.fromAssetId(1013524);
  const airplaneEntity = viewer.entities.add({
```

14. Mengamati hasilnya.


**Task 3**

1. Membuat *project baru glitch*, kemudian membuka *file index.html.*

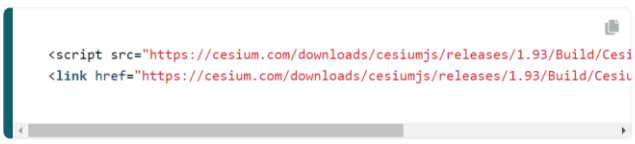



2. Menyalin token ke dalam *script.*

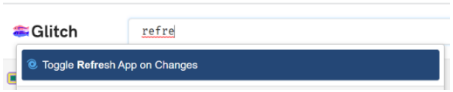
```
<script>
  // Get your token from https://cesium.com/ion/tokens
  Cesium.Ion.defaultAccessToken = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
  const viewer = new Cesium.Viewer('cesiumContainer');
```

3. Mencari *tool show app.*


4. Menyalin *script css* dan *js* pada *cesium.*



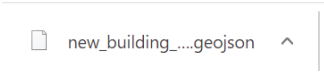
5. Menambahkan *refresh app*.



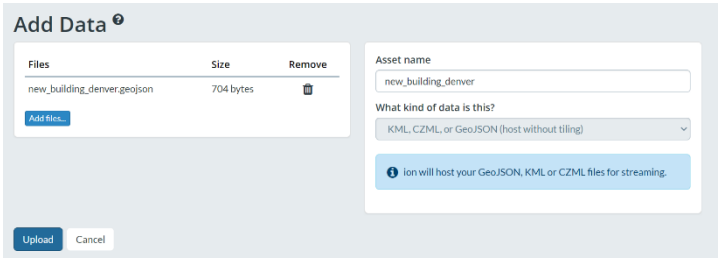

6. Menyalin script seperti berikut ini untuk menambahkan *OSM Buildings* dan *Cesium Terrain.*

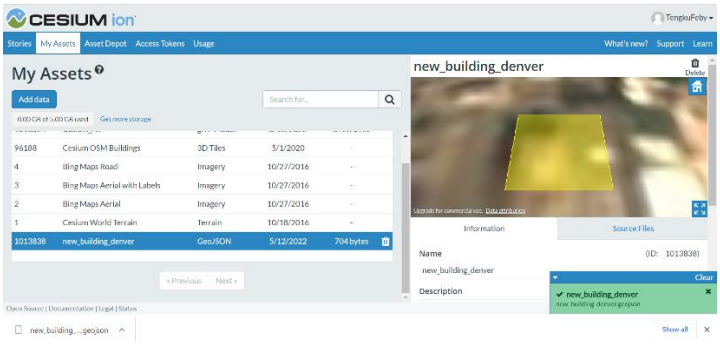
```
// Keep your Cesium.Ion.defaultAccessToken = 'your_token_here' line above.
// STEP 2 CODE
// Initialize the viewer with Cesium World Terrain.
const viewer = new Cesium.Viewer('cesiumContainer', {
  terrainProvider: Cesium.createWorldTerrain()
});
// Add Cesium OSM Buildings.
const buildingsTileset = viewer.scene.primitives.add(Cesium.createOsmBuildi
// Fly the camera to Denver, Colorado at the given longitude, latitude, anc
viewer.camera.flyTo({
  destination: Cesium.Cartesian3.fromDegrees(-104.9965, 39.74248, 4000)
});
```

7. Melakukan *download file GeoJSON.*



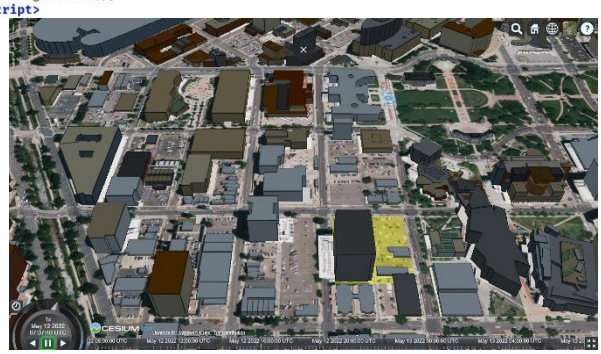

8. Membuka *cesiun ion dashboard*, kemudian melakukan *drag file* yang telah di *download*, dan klik *download.*



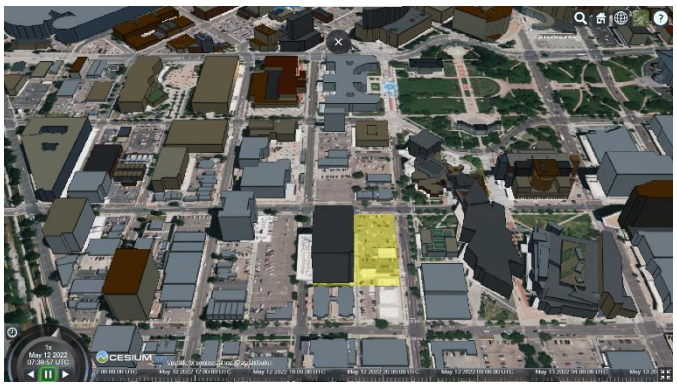
9. Menuliskan *script* seperti beriku ini, kemudian menyalin *ID*.

```
    // STEP 3 CODE
async function addBuildingGeoJSON() {
  // Load the GeoJSON file from Cesium ion.
  const geoJSONURL = await Cesium.IonResource.fromAssetId(1013838);
  // Create the geometry from the GeoJSON, and clamp it to the ground.
  const geoJSON = await Cesium.GeoJsonDataSource.load(geoJSONURL, { clampToGround: true });
  // Add it to the scene.
  const dataSource = await viewer.dataSources.add(geoJSON);
  // By default, polygons in CesiumJS will be draped over all 3D content in the scene.
  // Modify the polygons so that this draping only applies to the terrain, not 3D buildings.
  for (const entity of dataSource.entities.values) {
    entity.polygon.classificationType = Cesium.ClassificationType.TERRAIN;
  }
  // Move the camera so that the polygon is in view.
  viewer.flyTo(dataSource);
}
addBuildingGeoJSON();
  </script>
```



10. Menuliskan *script* seperti berikut ini untuk menghilangkan bangunan-bangunan kecil.

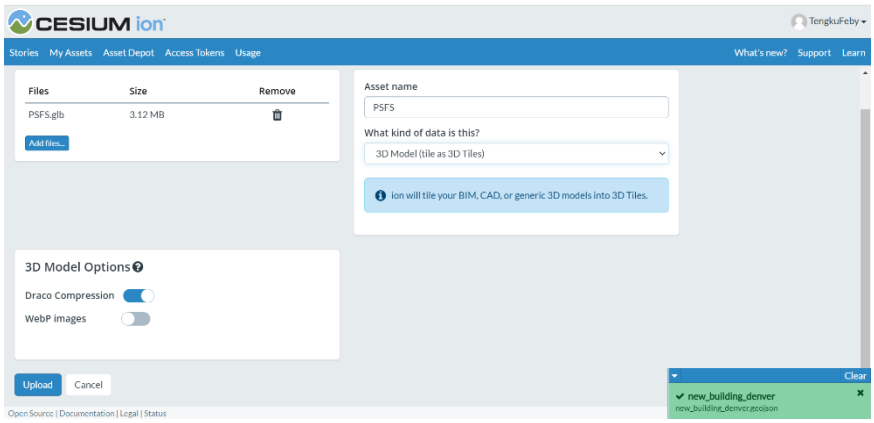
```
    // STEP 4 CODE
// Hide individual buildings in this area using 3D Tiles Styling language.
buildingsTileset.style = new Cesium.Cesium3DTileStyle({
  // Create a style rule to control each building's "show" property.
  show: {
    conditions : [
      // Any building that has this elementId will have `show = false`.
      ['${elementId} === 332469316', false],
      ['${elementId} === 332469317', false],
      ['${elementId} === 235368665', false],
      ['${elementId} === 530288180', false],
      ['${elementId} === 530288179', false],
      // If a building does not have one of these elementIds, set `show = true`.
      [true, true]
    ]
  },
  // Set the default color style for this particular 3D Tileset.
  // For any building that has a `cesium#color` property, use that color, otherwise make it wh
  color: "Boolean(${feature['cesium#color']}) ? color(${feature['cesium#color']}) : color('#ff
});
```



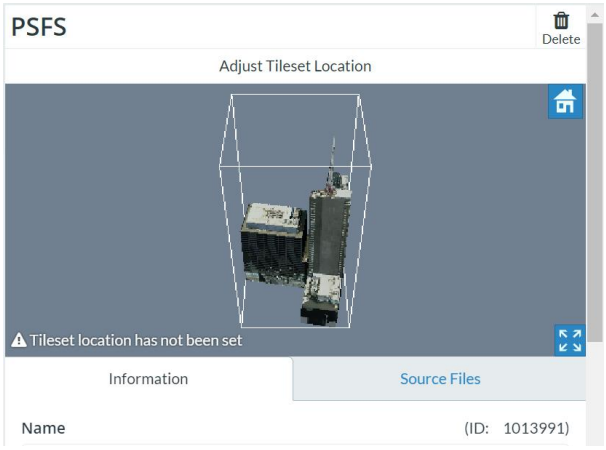
11. Mengunduh *Gtif Model.*

**1.** Download this glTF model.

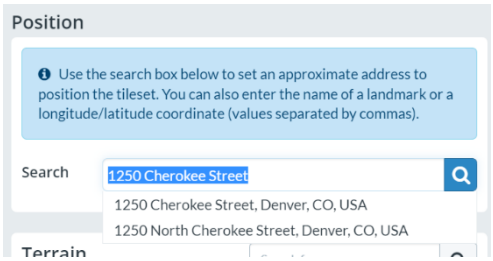
12. Melakukan *drag and drop* hasil *file* yang telah di unduh, kemudian memilih *3D Model (tile as 3D Tiles)*, kemudian menekan *Upload.*



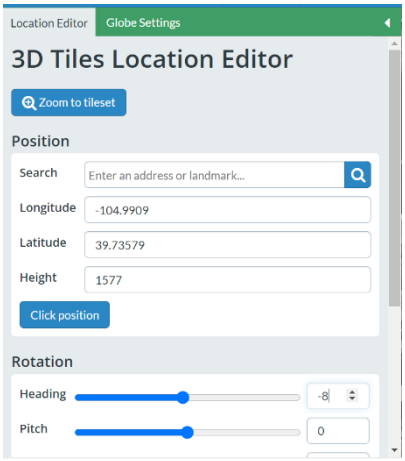

13. Melakukan klik *adjust tilest location.*




14. Mengetikkan alamat *1250 Cherokee Street* pada *search box* dan melakukan klik *next.*

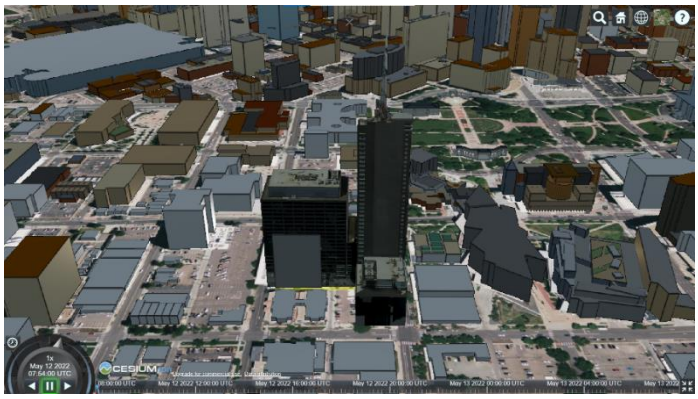


15. Mengedit *longitude*, *latitude*, *heading*, dan *height*, kemudian menekan *save.*




16. Menuliskan *script* seperti berikut ini disertain dengan *ID* untuk menambahkan bangunan pada *scene*.

```
    // STEP 6 CODE
// Add the 3D Tileset you created from your Cesium ion account.
const newBuildingTileset = viewer.scene.primitives.add(
  new Cesium.Cesium3DTileset({
    url: Cesium.IonResource.fromAssetId(1013991)
  })
);
// Move the camera to the new building.
viewer.flyTo(newBuildingTileset);
```



17. Menuliskan *script* seperti berikut ini pada <*body*> untuk menuju pada bagunan baru.

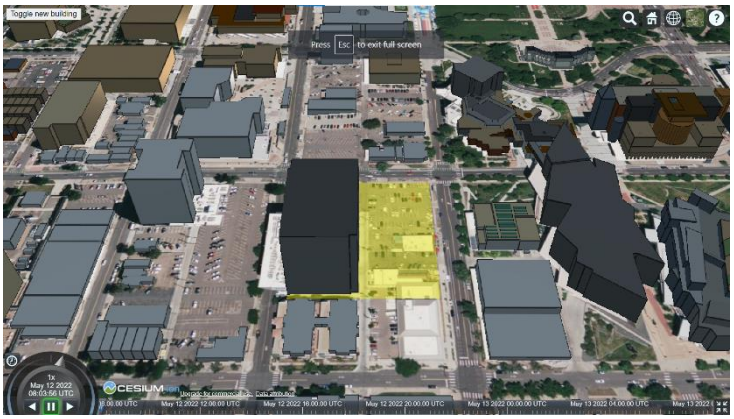
```
  </script>
</body>
  <button id="toggle-building">Toggle new building</button>
```

18. Menuliskan *script* seperti berikut ini untuk menambahkan *css style.*

```
<link href="style.css" rel="stylesheet">
  <style type="text/css">
  #toggle-building { z-index: 1; position: fixed; top: 5px; left: 5px; }
</style>
```

19. Menambahkan *scipt* kembali untuk mengaktifkan *button toggle*, kemudian mengamati hasil muncul dan tidaknya Gedung ketika di klik.

```
// STEP 7 CODE
// Toggle the tileset's show property when the button is clicked.
document.querySelector('#toggle-building').onclick = function() {
  newBuildingTileset.show = !newBuildingTileset.show;
};
```



## Task 4

1. Membuka *Cesium and Castle* dengan *basic style*.

JavaScript code | HTML body & CSS

```
// Grant CesiumJS access to your ion assets
// Cesium.Ion.defaultAccessToken = "_your_cesium_ion_acess_token_";

var viewer = new Cesium.Viewer("cesiumContainer", {
  terrainProvider: Cesium.createWorldTerrain(),
  animation: false,
});

// Adjust the camera to look at JAKPUS
viewer.camera.lookAt(
  new Cesium.Cartesian3.fromDegrees(106.8192454001557, -6.170262304004404),
  new Cesium.Cartesian3(0.0, -1500.0, 1200.0)
);

// Add OSM Building tileset
var osmBuildingsTileset = Cesium.createOsmBuildings();
```


2. Mengaturan nama fitur berdasarkan keterangan bangunan.

```
// Pengaturan fitur based on property
function showFeaturesWithProperty() {
  osmBuildingsTileset.style = new Cesium.Cesium3DTileStyle({
  show: "${feature['building']} === 'Office'",
});
}
```

3. Menambahkan *script* untuk memberikan visualisasi warna sesuai jarak antar bangunan.

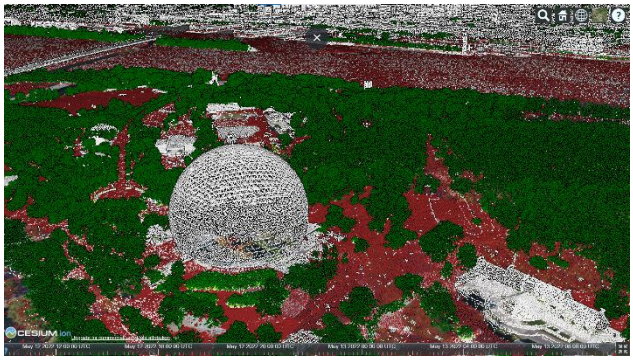
```
// Color features with conditions
function colorFeaturesWithConditions() {
  osmBuildingsTileset.style = new Cesium.Cesium3DTileStyle({
  defines: {
    distanceFromComplex:
      "distance(vec2(${feature['cesium#longitude']}, ${feature['cesium#latitude']}), vec2(106.8192454001
  },
  color: {
    conditions: [
      ["${distanceFromComplex} > 0.010", "color('#d65c5c')"],
      ["${distanceFromComplex} > 0.006", "color('#f58971')"],
      ["${distanceFromComplex} > 0.002", "color('#f5af71')"],
      ["${distanceFromComplex} > 0.0001", "color('#f5ec71')"],
      ["true", "color('#ffffff')"],
    ],
  },
});
}

```

4. Menuliskan *script* seperti berikut ini untuk membuat *point cloud.*

JavaScript code | HTML body & CSS

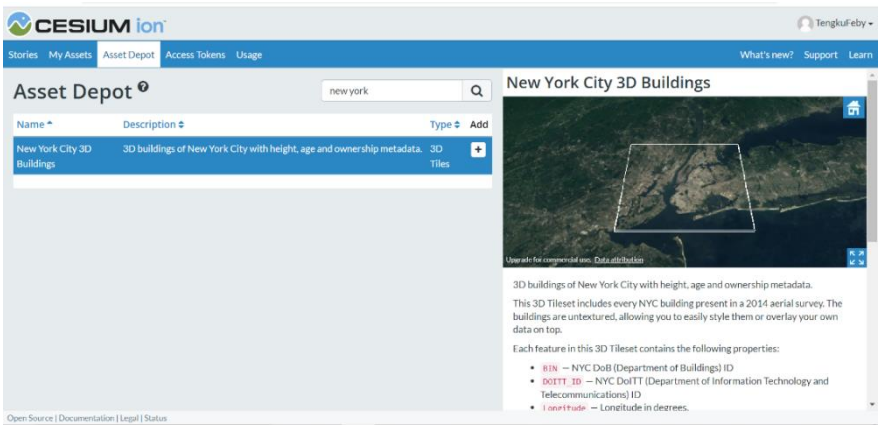
```
var viewer = new Cesium.Viewer("cesiumContainer", {
  terrainProvider: Cesium.createWorldTerrain(),
  animation: false
});

// A ~10 billion point 3D Tileset of the city of Montreal, Canada captured in 201
var pointCloudTileset = viewer.scene.primitives.add(
  new Cesium.Cesium3DTileset({
    url: Cesium.IonResource.fromAssetId(28945),
    pointCloudShading: {
      attenuation: true,
      maximumAttenuation: 2,
    },
  })
);

// Fly to the Biosphere Museum
viewer.camera.setView({
  destination: new Cesium.Cartesian3(
    1269319.8408991008,
    -4293301.826913256,
    4527724.561372451
  ),
  orientation: {
    direction: new Cesium.Cartesian3(
      -0.74250503010783 2,
```

Gallery | Console

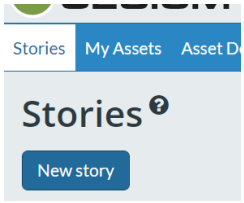


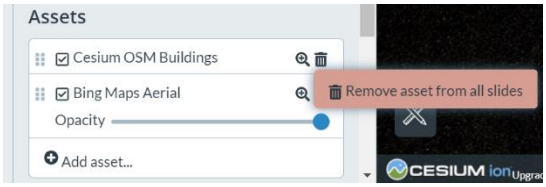
**Melakukan *editing style* pada *Cesium Stories***

1. Membuka https://cesium.com/ion.
2. Membuka *Asset Depot*, kemudian memilih *New York City Building*, dan menekan *add to my asset.*

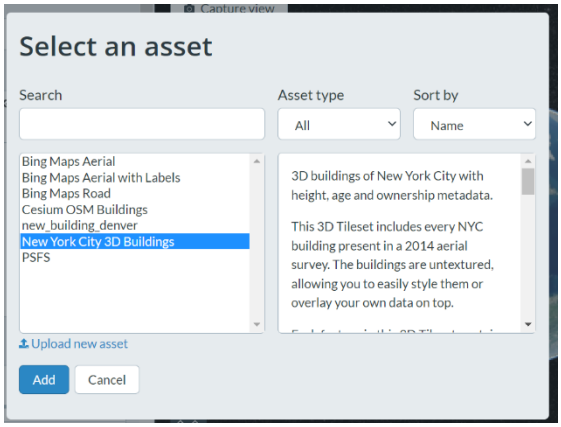


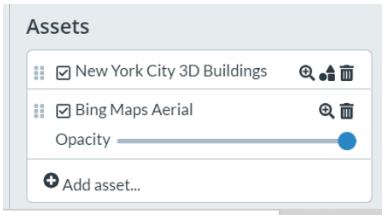
3. Melakukan *add data* dengan menekan *button new stories*, kemudian menghapus *Cesium OSM Building* untuk menampilkan bangunan di New Yok saja.



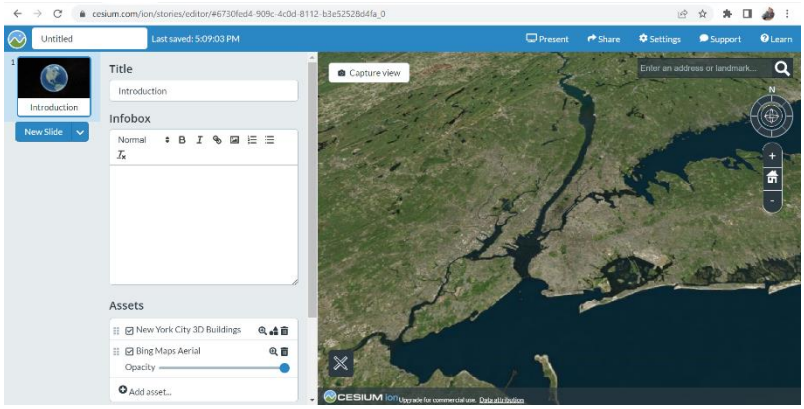




4. Menambahkan *asset New Yok City Building* yang sebelumnya telah ditambahkan.

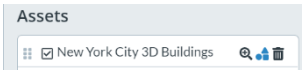


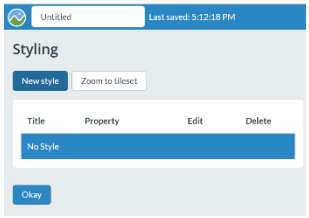



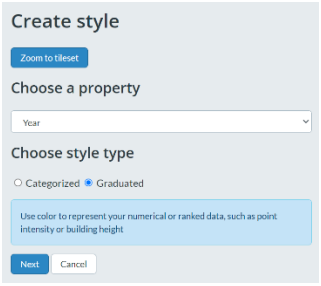

5. Melakukan *Zoom In* pada area kajian New York.



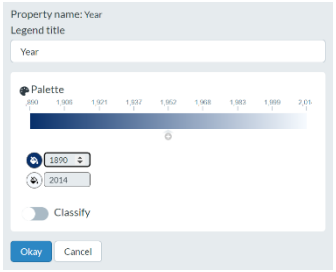
6. Membuat *style* baru dengan cara menekan *button style*, kemudian melakukan pengaturan lainnya.

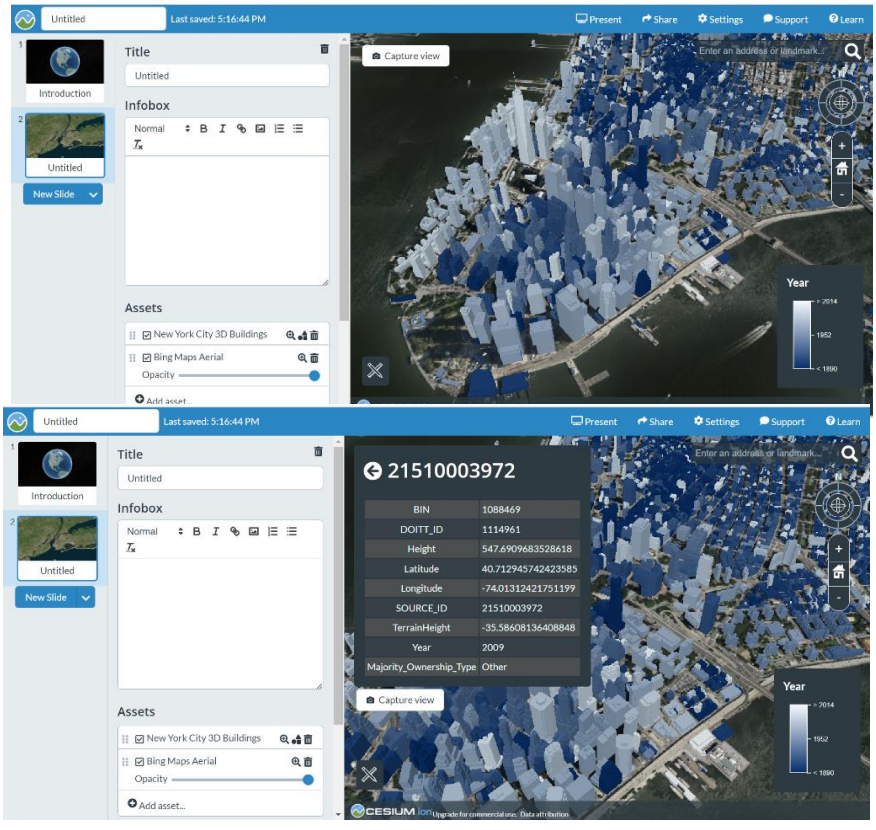


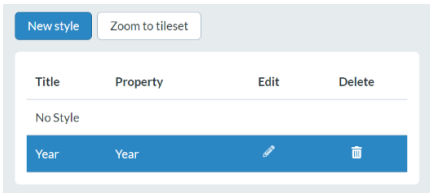



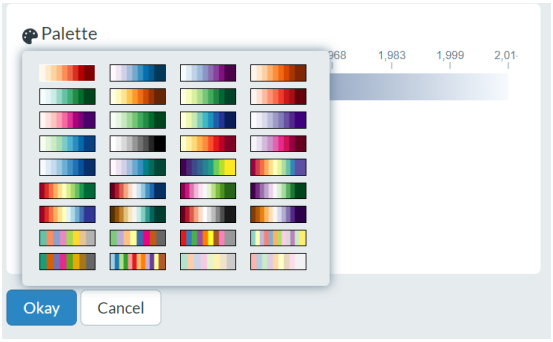

7. Melakukan *styling* pada Gedung tahun 2014 dan 1890, dengan melakukan pengaturan seperti berikut ini, lalu menekan *button okay*.

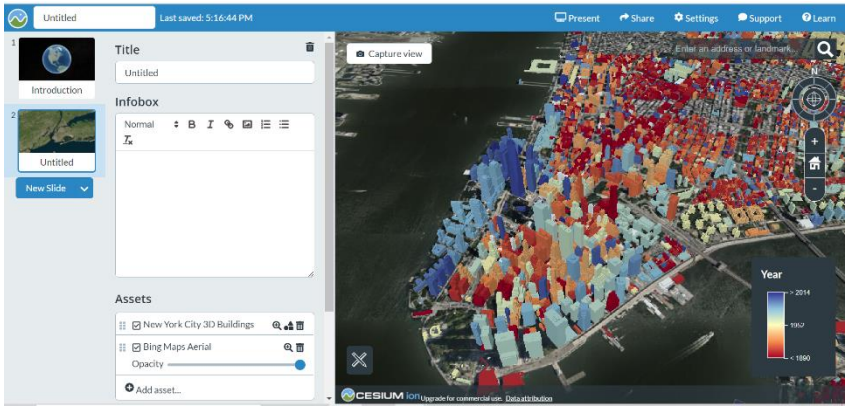


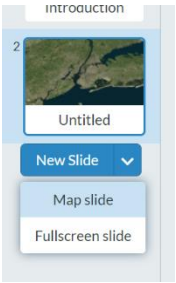
8. Melakukan *styling* kembali.

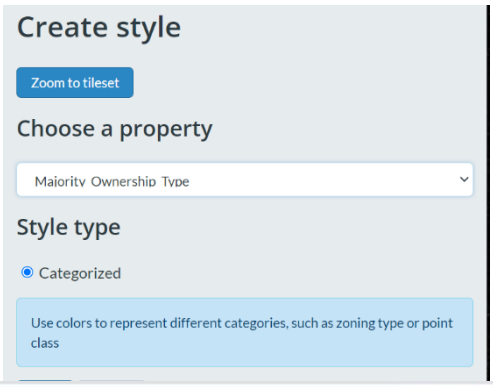


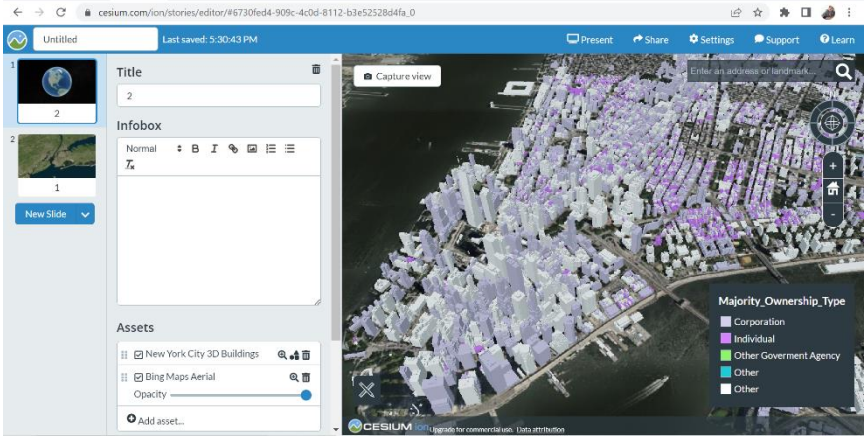



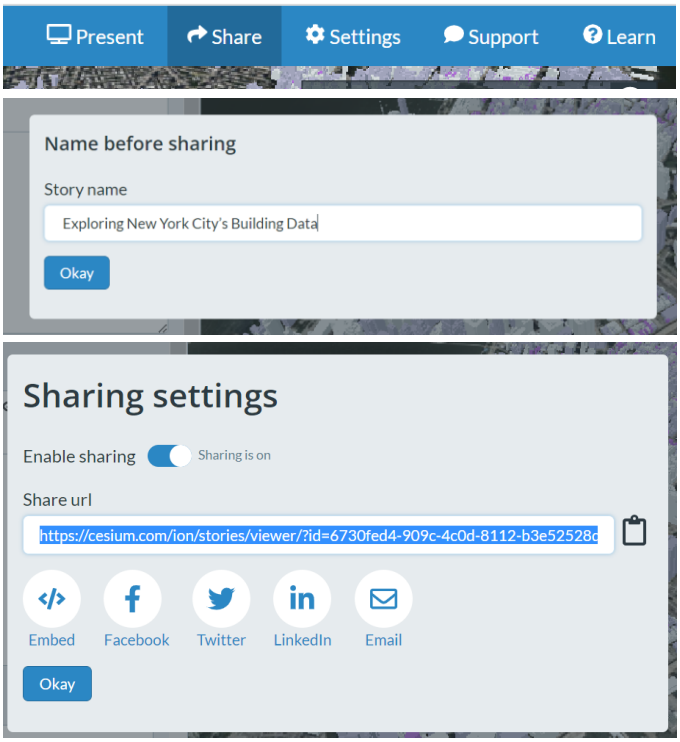

9. Melakukan *style* untuk kategorisasi data, dan klik *save*.



10. Melakukan *share story*.

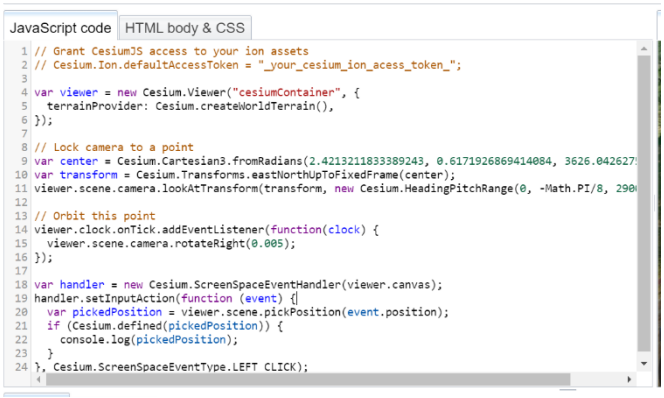


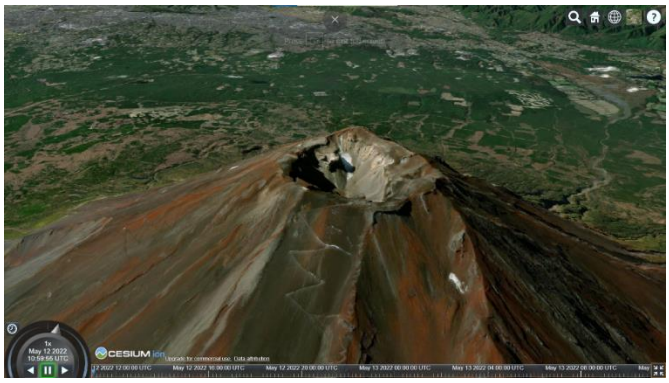

*Share url*

| |
|---|
| https://cesium.com/ion/stories/viewer/?id=6730fed4-909c-4c0d-8112-b3e52528d4fa |

**Task 5**

1. Membuka *cesium and sandcastle*.
2. Mengetikkan *script* seperti berikut ini.

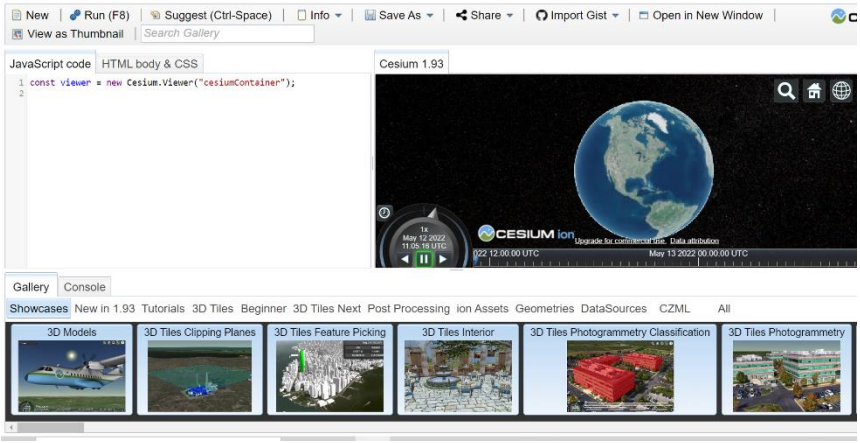




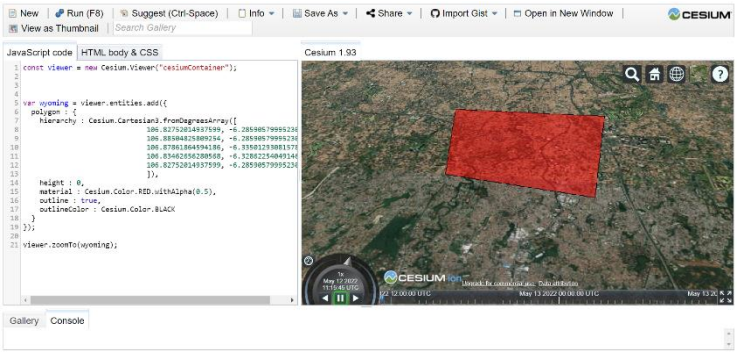
**Task 6**

1. Membuka *Cesium & Sand Castle*.

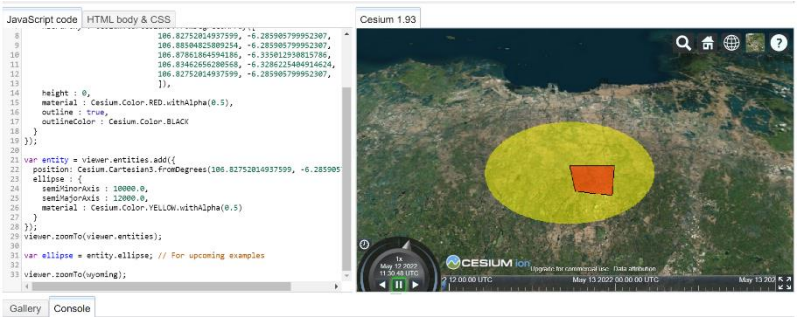
   https://sandcastle.cesium.com/?src=Hello%20World.html&label=Showcases.



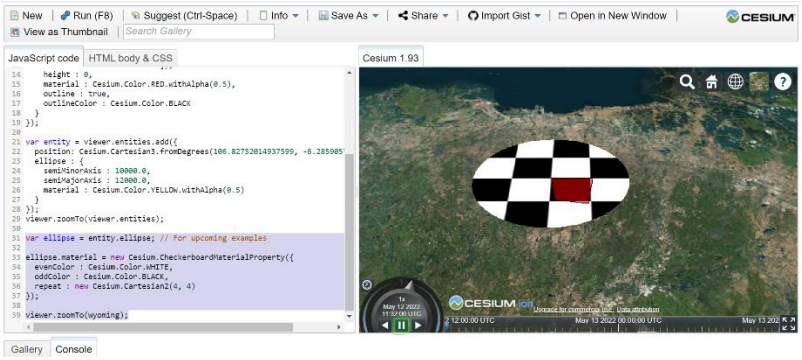

2. Menambahkan *polygon* pada area kajian.



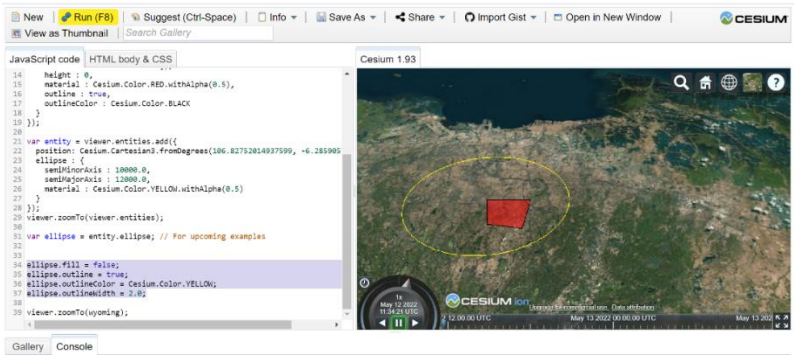
3. Menuliskan *script* untuk menampilkan elips.




4. Menuliskan *script* untuk menampilkan *clipboard.*




5. Menuliskan *script* seperti berikut ini untuk menampilkan *outline elips.*




**Task 7**

**Pencahayaan**

1. Membuka *visual studio*, kemudian memasukkan *CesiumJS JavaScript* dan *CSS files.*

```
<head>
  <meta charset="utf-8">
  <!-- Include the CesiumJS JavaScript and CSS files -->
  <script src="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Cesium.js"></script>
  <link href="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Widgets/widgets.css" rel="stylesheet">
</head>
```

2. Menyalin *token* pada *script.*

```
<body>
  <div id="cesiumContainer"></div>
  <script>
    // Your access token can be found at: https://cesium.com/ion/tokens.
    // Replace `your_access_token` with your Cesium ion access token.

    Cesium.Ion.defaultAccessToken = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI0NDA1MGExNS00NDgyLTRkNDUtOTFlNy1hMjJjMjQxOTU5YjYiLCJpZ
```

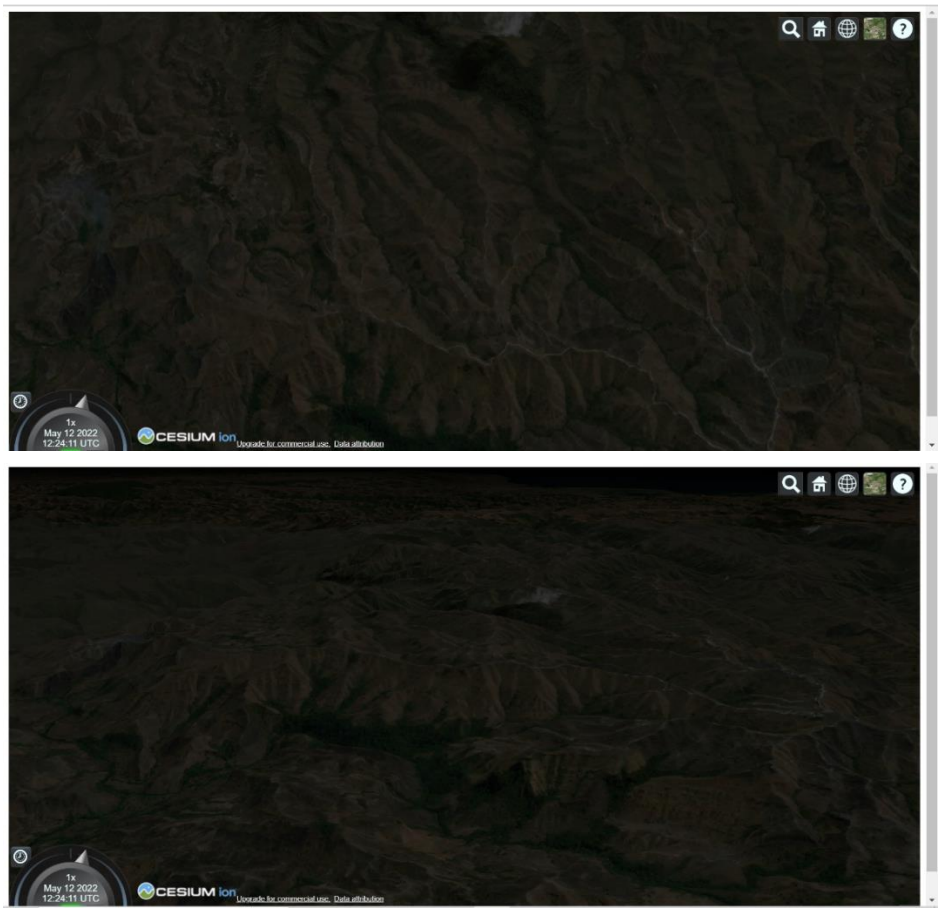
3. Menulsikan *script* untuk menampilkan efek *terrain lighting.*

```
// Initialize the Cesium Viewer in the HTML element with the `cesiumContainer` ID.
var viewer = new Cesium.Viewer('cesiumContainer', {
  terrainProvider: Cesium.createWorldTerrain({
    requestVertexNormals: true
  })
});
viewer.scene.globe.enableLighting = true;
```

4. Menuliskan *script* seperti berikut ini untuk mengarahkan kamera ke area kajian yakni Pulau Sumba.

```
// Fly the camera to Pulau Sumba at the given longitude, latitude, and height.
viewer.camera.flyTo({
  destination: Cesium.Cartesian3.fromDegrees(120.29449429789291, -9.903764361255192, 10000),
orientation : {
    heading : Cesium.Math.toRadians(0.0),
    pitch : Cesium.Math.toRadians(-70.0),
  }
});
```

5. Mengamati hasilnya.



**Efek Air**

1. Membuka visual studio, kemudian memasukkan *CesiumJS JavaScript* dan *CSS files.*

```
<head>
  <meta charset="utf-8">
  <!-- Include the CesiumJS JavaScript and CSS files -->
  <script src="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Cesium.js"></script>
  <link href="https://cesium.com/downloads/cesiumjs/releases/1.93/Build/Cesium/Widgets/widgets.css" rel="stylesheet">
</head>
```

2. Menyalin token pada *script.*

```
<body>
  <div id="cesiumContainer"></div>
  <script>
    // Your access token can be found at: https://cesium.com/ion/tokens.
    // Replace `your_access_token` with your Cesium ion access token.

    Cesium.Ion.defaultAccessToken = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI0NDA1MGExNS00NDgyLTRkNDUtOTFlNy1hMjJjMjQxOTU5YjYiLCJp
```

3. Menulsikan *script* untuk menampilkan efek *Watermask.*

```
// Initialize the Cesium Viewer in the HTML element with the `cesiumContainer` ID.
var viewer = new Cesium.Viewer('cesiumContainer', {
  terrainProvider: Cesium.createWorldTerrain({
    requestWaterMask: true
  })
});
```

4. Menuliskan *script* seperti berikut ini untuk mengarahkan kamera ke area kajian yakni Lapaoe Pulau Sumba.

```
// Fly the camera to Lapaoe Pulau Sumba at the given longitude, latitude, and height.
viewer.camera.flyTo({
  destination: Cesium.Cartesian3.fromDegrees(120.02594964869141, -9.383740861729862, 10000),
  orientation: {
    heading: Cesium.Math.toRadians(0.0),
    pitch: Cesium.Math.toRadians(-70.0),
  }
});
```