esri Indonesia
THE SCIENCE OF WHERE

Location Intelligence
Site Selection: Restaurant Cluster

edu-smartcommunity.hub.arcgis.com

Muhammad Zayyanul Afwani
education@esriindonesia.co.id

Klasterisasi Restoran

Site selection pada awalnya merupakan strategi untuk mengembangkan suatu bisnis berdasarkan potensi suatu wilayah. Pertumbuhan pembangunan yang sangat pesat yang didukung dengan jumlah populasi disuatu daerah membuat adanya fluktuasi pasar terutama dibidang *retail.*

Integrasi penentuan lokasi atau *site selection* dengan GIS akan mempermudah pelaku bisnis untuk memperluas usaha mereka dengan cara membuka market baru. Dalam studi kasus ini mengangkat penentuan lokasi untuk menentukan potensi pengembangann bisnis restoran. Penentuan lokasi restoran yang tepat tentunya sangat penting untuk menunjang bisnis kedepannya, sehingga pelaku bisnis perlu menentukan lokasi yang potensial. Salah satu cara untuk menentukan lokasi potensial ini yaitu menggunakan data historis seperti tempat yang yang sering dikunjungi untuk belanja, makan, atau bersantai sejenak.

Pada sesi ini kita akan melihat bagaimana Informasi Geospasial dapat melakukan klasterisasi restoran yang memiliki potensial . Dengan berperan sebagai tim pengembang aplikasi, setidaknya kita dapat membantu memasarkan produk untuk mengurangi pencemaran limbah di lingkungan kita.

Estimasi waktu : 60 Menit

Pada latihan ini Anda membutuhkan:

Akun ArcGIS Online organisasi

ArcGIS Pro

Informasi mengenai penggunaan LearnArcGIS selama masa Pandemi

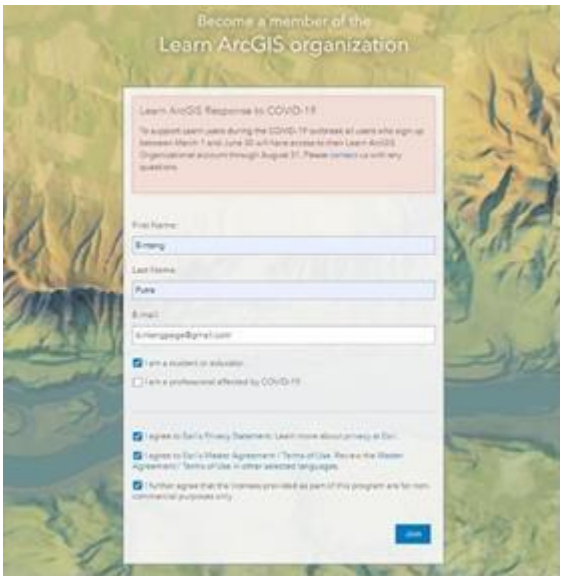
Untuk membantu pekerja yang terimbas dalam penanggulangan COVID19 : Esri memberikan akses pembelajaran gratis ArcGIS Online dan lebih dari 20 apps sampai 21 Januari 2020, termasuk ArcGIS Pro, dan beberapa course dalam Learn ArcGIS dan Esri Academy. Untuk mendapatkan akses silahkan mendaftar di https://learn.arcgis.com/en/become-a-member/

esri Indonesia

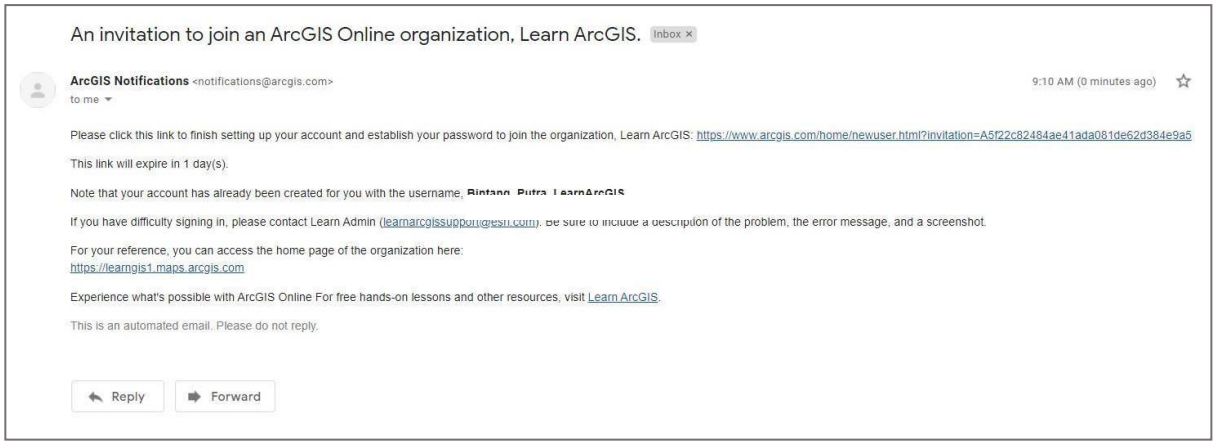
Cara Registrasi Akun LearnArcGIS

1. Akses link melalui https://learn.arcgis.com/en/become-a-member/

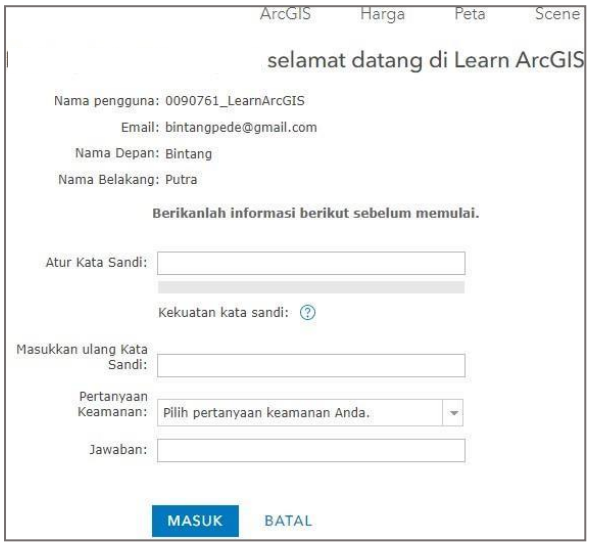
2. Isi data diri yang diminta, kemudian klik join



3. Cek Inbox E-mail yang didaftarkan pada Learn ArcGIS.

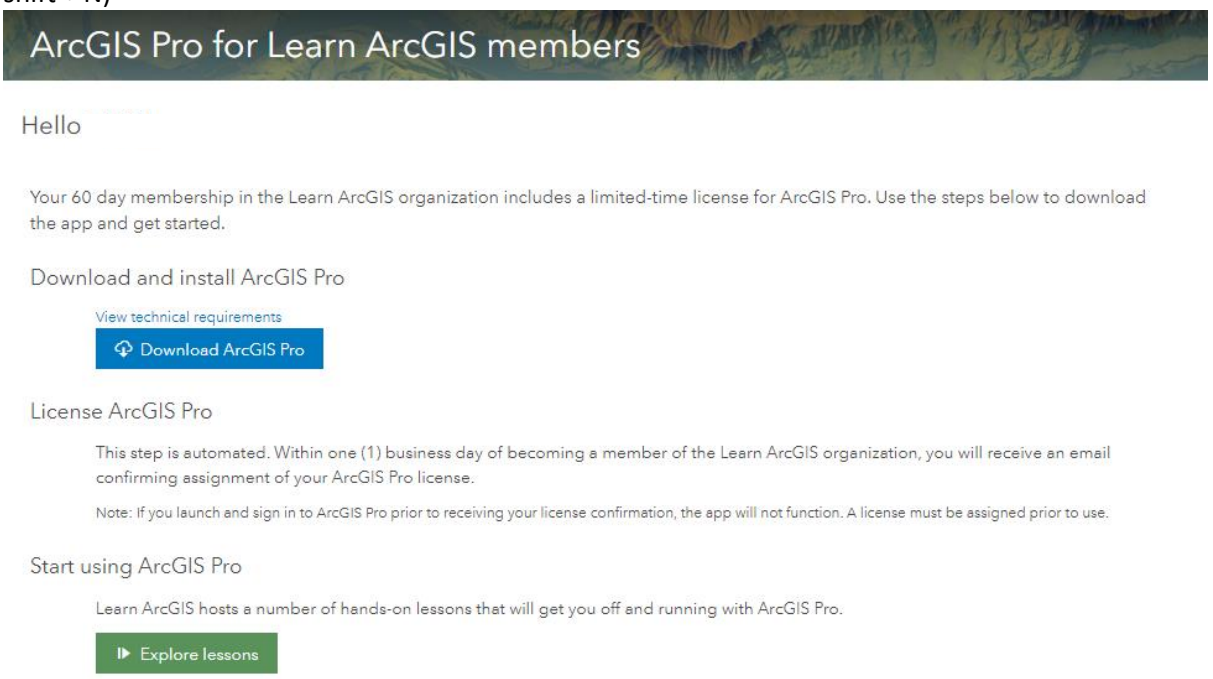


4. Isi sandi beserta pertanyaan keamanan yang diperlukan

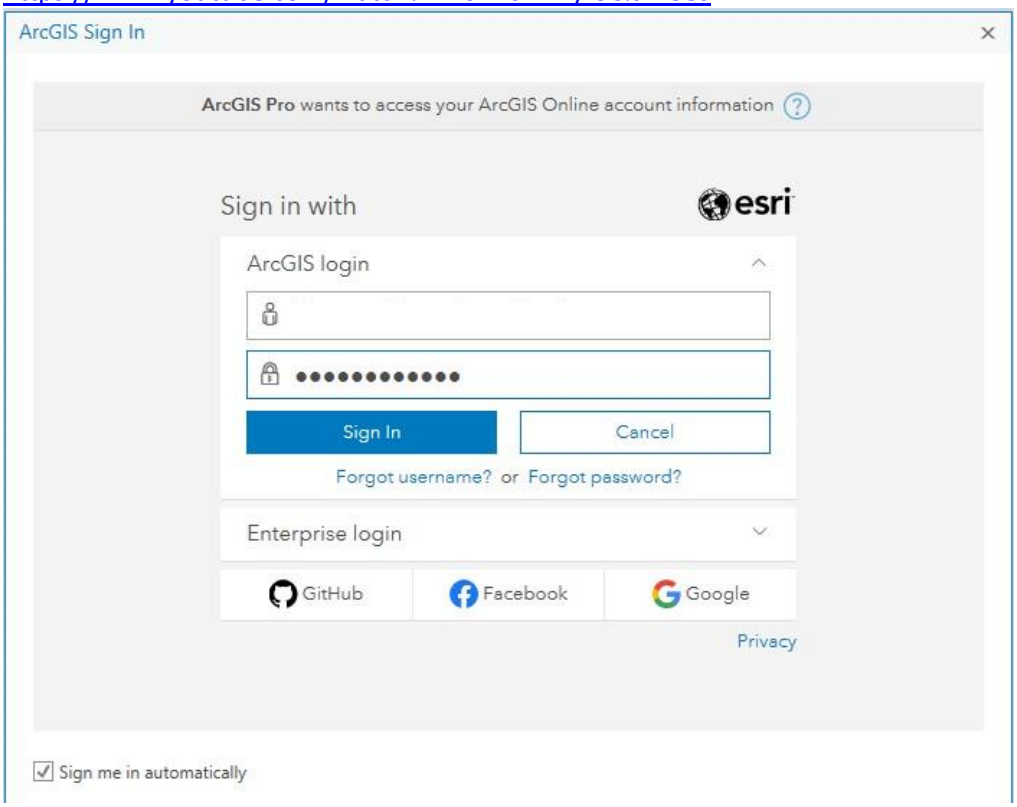


5. Selamat, anda sudah mendapatkan free subscription ArcGIS Online selama masa pandemik

1. Akses link di http://tiny.cc/unduhgis (Jika tidak bisa dibuka di browser biasa, unduh di incognito browser dengan shift + N)



2. Ikuti Langkah-langkah unduh dan penginstallan, kemudian jika ingin masuk sign in menggunakan akun yang sama https://www.youtube.com/watch?v=k0H_s1IWyiU&t=438s



Step 1 : Login ke Akun ArcGIS Online

1. Buka browser dan akses https://www.arcgis.com/

   ArcGIS online adalah website dinamis, tampilan yang muncul dapat berbeda dari gambar yang terlihat pada modul.

Klik sign in, kemudian ketik username dan password Anda.

Bagi yang tidak memiliki username dan password dapat menggunakan username learn https://learn.arcgis.com/en/become-a-member/ dengan mengikuti langkah-langkah diatas, atau menggunakan akun contoh :

Username : 0124538_LearnArcGIS

Password : esriindonesia2020

Username : UNGorontalo_Workshop_LearnArcGIS

Password : esriindonesia2020

Username : UNG_Workshop_LearnArcGIS

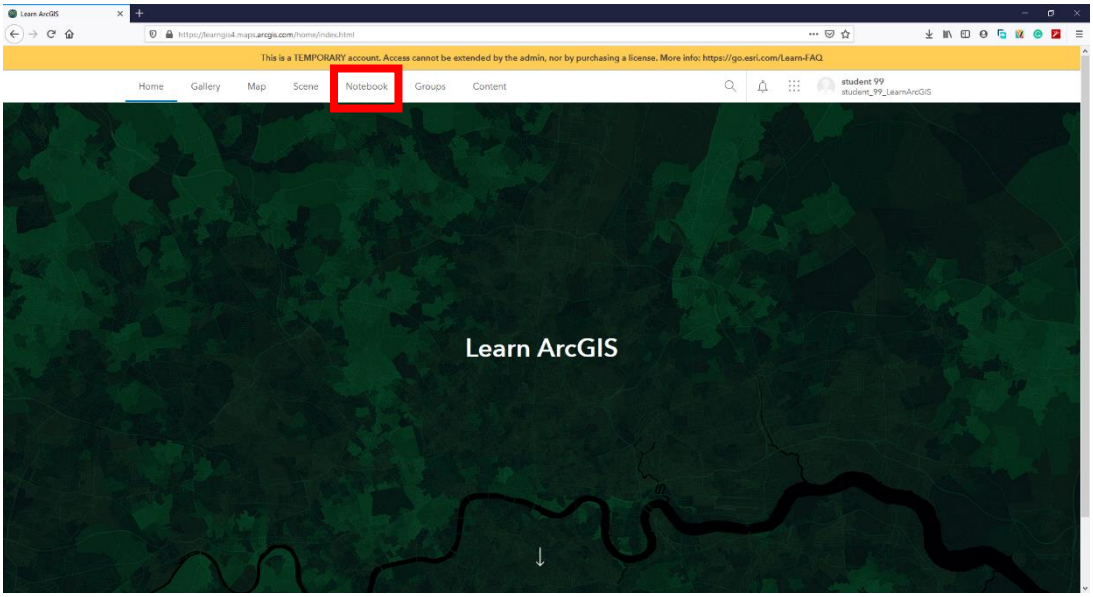
Password : esriindonesia2020

Step 2: Membuka ArcGIS Notebook

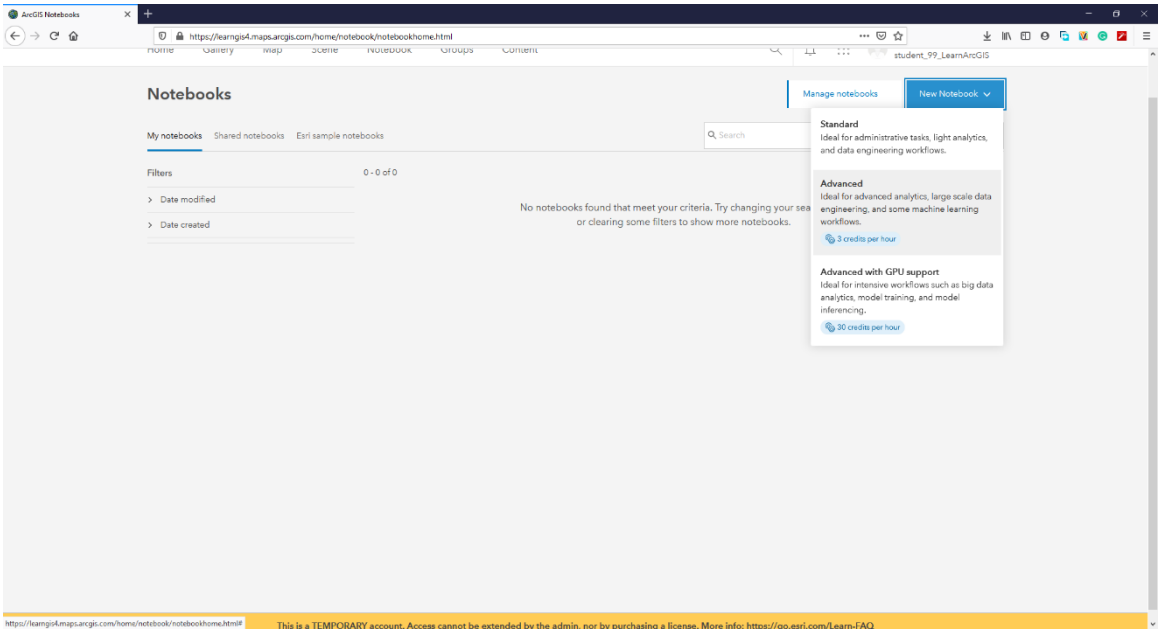
1. Penjelasan mengenai ArcGIS Notebook

ArcGIS Notebook memiliki tampilan yang sama dengan jupyter notebook dengan kata lain ArcGIS Notebook memiliki *Jupyter notebook environment*. Seluruh environment dihosting di ArcGIS Enterprise dan diberdayakan oleh ArcGIS Notebook Server. ArcGIS Notebook memiliki kemampuan untuk melakukan *Big Data Analysis, Machine Learning, Deep Learning* dan Visualisasi data geospasial.

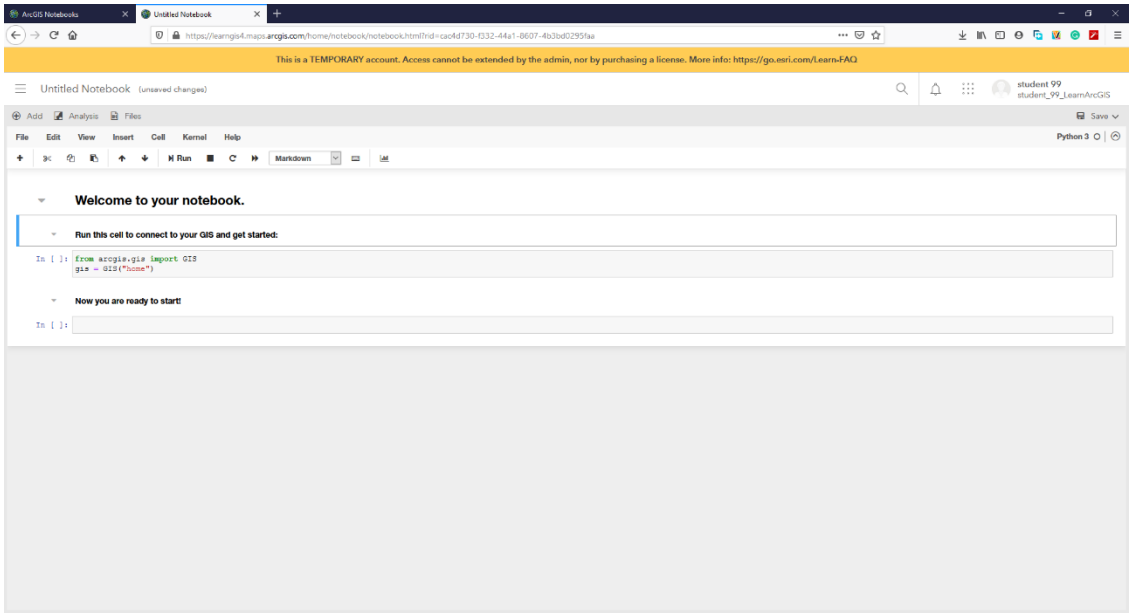
Untuk membuka ArcGIS Notebook maka klik Notebook pada tampilan utama ArcGIS Online Anda



2. Buatlah Notebook baru dengan cara mengklik “New Notebook” kemudian pilih “Advanced”. Untuk melakukan pemrosesan Deep Learning atau Big Data Analytics yang memerlukan GPU, Anda bisa menggunakan Advanced with GPU support. Namun, hal itu akan memakan banyak credit.



3. Tampilan ArcGIS Notebook akan seperti gambar dibawah. Selanjutnya mulailah menulis script Anda pada notebook ini.



Step 3: Import module

1. Import module yang dibutuhkan ke ArcGIS Notebook, Anda tidak perlu melakukan instalasi library melalui environment python di local machine Anda.

```
import os
import time

import numpy as np
import pandas as pd
# Imports for plotting in Bokeh
import bokeh
from bokeh.io import output_notebook, output_file, show
from bokeh.plotting import figure
from bokeh.models import Legend, Range1d
from bokeh.embed import file_html
from bokeh.resources import CDN
# Set bokeh to output plots in the notebook
output_notebook()

import arcgis
from arcgis import features
from arcgis.features import GeoAccessor, GeoSeriesAccessor
from arcgis.geoenrichment import *
from arcgis.geometry import project
import arcpy
arcpy.env.overwriteOutput = True

gis = arcgis.gis.GIS("home", verify_cert=False)
```

2. Selanjutnya buatlah fungsi berikut yang mendefinisikan spatial-enabled dataframe yang digunakan untuk mengeglola basis data spasial

```
def create_map_and_histogram(map_location, sdf, column,
                             method='esriClassifyNaturalBreaks',
                             class_count=5, cmap='OrRd',
                             alpha=0.8, plot_height=600, plot_width=600):
    # Create map
    map_obj = gis.map(map_location)
    sdf.spatial.plot(map_widget=map_obj, renderer_type='c',
                     method=method, class_count=class_count,
                     col=column, cmap=cmap, alpha=0.8)

    # Extract the layer's class breaks and colors
    class_breaks = map_obj.layers[0].layer\
                      .layerDefinition.drawingInfo.renderer\
                      .classBreakInfos
    cbs_list = []
    cmap_list = []
    for cb in class_breaks:
        cbs_list.append(cb.classMaxValue)
        cmap_list.append('#%02x%02x%02x' % (cb.symbol.color[0],
                                            cb.symbol.color[1],
                                            cb.symbol.color[2]))

    # Create a histogram of salesvol values
    hist, edges = np.histogram(sdf[column],
                               bins=class_count)

    # Put the information in a dataframe
    hist_df = pd.DataFrame({column: hist,
                            'left': edges[:-1],
                            'right': edges[1:]})

    # Add colors to each hist_df record
    hist_df['color'] = pd.Series(cmap_list)

    # Create the blank plot
    p = figure(plot_height = plot_height, plot_width = plot_width,
               title = 'Histogram',
               y_axis_label = 'Feature Count',
               x_axis_label = column)

    # Add a quad glyph
    p.quad(bottom=0, top=hist_df[column],
           left=hist_df['left'], right=hist_df['right'],
           line_color='white', fill_color=hist_df['color'])

    # Return outputs
    return map_obj, p
```

3. Mengganti extent peta

```
def return_center_dict(centerx, centery):
    center_shape_dict = {}
    center_shape_dict['spatialReference'] = \
        {'latestWkid': 3857, 'wkid': 102100}
    center_shape_dict['x'] = centerx
    center_shape_dict['y'] = centery
    return center_shape_dict

def visit_sites(m, points, timesleep, zoom_level=15):
    m.zoom = zoom_level
    i = 0
    for i in range(len(points)):
        m.center = [points[i]['y'], points[i]['x']]
        time.sleep(timesleep)

def visit_ellipse_features_on_map_widget(map_obj, sdf, zoom_level=15,
                                         number_of_records=None,
                                         timesleep=5):
    # Create a center_xy column from the sdf
    sdf['center_xy'] = sdf.apply(
        lambda x: return_center_dict(x['centerx'], x['centery']), axis=1)
    if number_of_records:
        sitelist = sdf.head(number_of_records)['center_xy'].tolist()
    else:
        sitelist = sdf.head()['center_xy'].tolist()
    xys = project(sitelist, in_sr=3857, out_sr=4326)
    visit_sites(map_obj, xys, timesleep)
```

Step 4: Retrieve Pittsburgh restaurant data

1. Esri menyediakan layanan yang dapat diakses secara gratis melalui layanan Living Atlas, Services dsb. Kali ini Anda akan mengambil salah satu layanan tersebut dan dihosting di akun ArcGIS Online Anda.

```
In [8]: data_groups = gis.groups.search('"ArcGIS Sample Notebooks Data" owner:esri_notebook',
                                      outside_org=True)
        group_query = f"group: {data_groups[0].id}" if data_groups else ""
        restaurants_item = gis.content.search(f'restaurants_pittsburgh {group_query}',
                                              item_type = "Feature Service",
                                              outside_org=True)[0]
        restaurants_item
```

Out[8]:

Feature Layer Collection by some_user
Last Modified: September 10, 2018
0 comments, 21 views

2. Spatial-enabled dataframe dapat pula digunakan untuk membuat geodatabase di akun ArcGIS Online Anda

```
restaurants_sdf = pd.DataFrame.spatial.from_layer(
    restaurants_item.layers[0])
restaurants_sdf.head()
```

```
# Create file geodatabase
fgdb = arcpy.CreateFileGDB_management(
    out_folder_path=home_dir,
    out_name="pittsburgh_restaurants.gdb").getOutput(0)
fgdb
```

```
'/mnt/arcgis/me/pittsburgh_restaurants.gdb'
```

```
# Use the spatially-enabled dataframe's to_featureclass
# method to convert to a feature class
fc = restaurants_sdf.spatial.to_featureclass(
    os.path.join(fgdb, "restaurants_pittsburgh"))
fc
```

```
'/mnt/arcgis/me/pittsburgh_restaurants.gdb/restaurants_pittsburgh'
```

```
# Confirm it exists after converting it
arcpy.Exists(fc)
```

```
True
```

Step 5: Explore data menggunakan table, diagram, dan peta

4. Mencari tahu berapa jumlah restoran dan kolom di data kita

```
# Get the amount of records and columns in our data
print("Amount of restaurants: {0}\nAmount of attributes: {1}".format(
          restaurants_sdf.shape[0], restaurants_sdf.shape[1]))
```

```
Amount of restaurants: 5026
Amount of attributes: 22
```

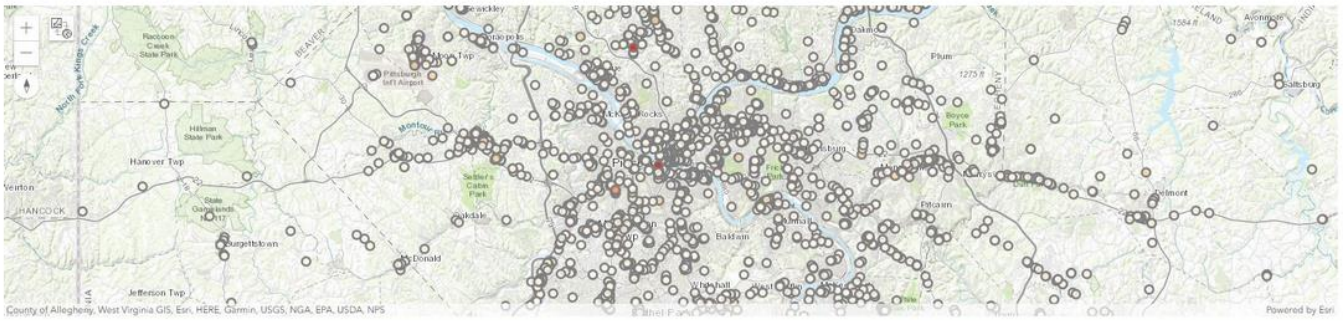
```
# Get summary statistics on numeric fields
restaurants_sdf.describe()
```

	empnum	objectid	salesvol	score
count	5026.000000	5026.000000	5026.000000	5026.000000
mean	17.216076	2513.500000	802.645444	99.608847
std	24.190230	1451.025557	1130.667265	1.855258
min	1.000000	1.000000	0.000000	85.703100
25%	6.000000	1257.250000	280.000000	100.000000
50%	8.000000	2513.500000	374.000000	100.000000
75%	19.000000	3769.750000	888.000000	100.000000
max	300.000000	5026.000000	14035.000000	100.000000

```
# Get a basic map of the restaurant locations as points
restaurants_map = gis.map("Pittsburgh")
restaurants_sdf.spatial.plot(map_widget=restaurants_map)
restaurants_map
```

5. Menggunakan helper function untuk menampilakn peta dan histogram berdasarkan “salesvol” atau volume penjualan

```
restaurants_map, sales_hist = create_map_and_histogram("Pittsburgh",
                                    restaurants_sdf, "salesvol")
restaurants_map
```




```
show(sales_hist)
```

6. Menentukan lima restoran dengan volume penjualan tertinggi

```
top_sales_sdf = restaurants_sdf.sort_values(
    by=['salesvol'],
    ascending=False).reset_index()
top_five_sdf = top_sales_sdf.head()

print("Top 5 Restaurants in Pittsburgh by Sales Volume: ")
top_five_sdf[['coname', 'salesvol', 'street', 'city', 'state',
              'state_name', 'zip']]
```

```
Top 5 Restaurants in Pittsburgh by Sales Volume:
```

	coname	salesvol	street	city	state	state_name	zip
0	CHEESECAKE FACTORY	14035.0	ROSS PARK MALL DR	PITTSBURGH	PA	Pennsylvania	15237
1	EAT'N PARK	12865.0	PARK MANOR DR	PITTSBURGH	PA	Pennsylvania	15205
2	CHEESECAKE FACTORY	12631.0	S 27TH ST	PITTSBURGH	PA	Pennsylvania	15203
3	M I FRIDAY	11695.0	PERRY HWY	PITTSBURGH	PA	Pennsylvania	15229
4	NANA	11695.0	WATERFRONT DR	PITTSBURGH	PA	Pennsylvania	15222

```
label = top_five_sdf['street'].tolist()
value = top_five_sdf['salesvol'].tolist()

p = figure(x_range=label,
           title = 'Top Five Restaurants by Sales Volume',
           x_axis_label = 'Restaurant Address',
           y_axis_label = 'Sales volume in thousands of dollars')
p.xaxis.major_label_orientation = np.pi/4

p.vbar(x=label, top=value, width=0.5)

show(p)
```

```
# Plot the highest five selling restaurants
top_five_restaurants_map = gis.map("Pittsburgh")
top_five_sdf.spatial.plot(map_widget=top_five_restaurants_map)
top_five_restaurants_map
```

7. Menentukan lima daerah berdasarkan zip code dengan volume penjualan tertinggi

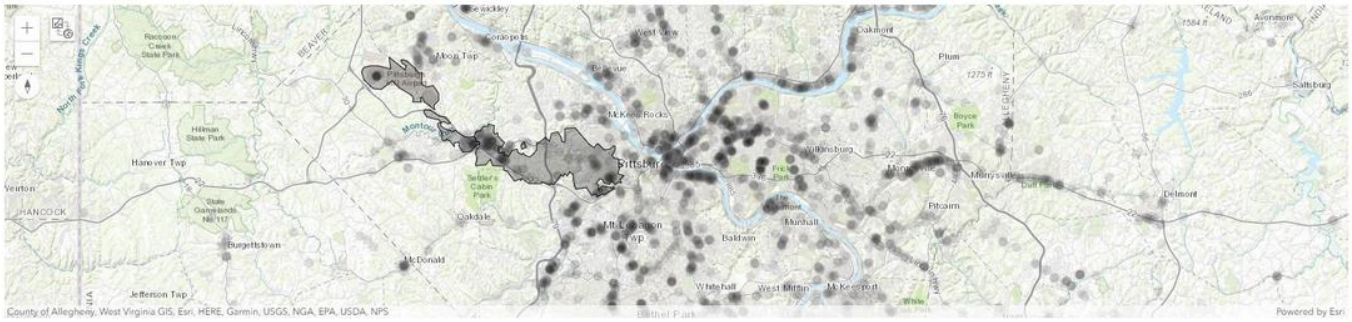
```
# Use groupby object to make calculations by Zip Code
zip_groupby = restaurants_sdf.groupby(restaurants_sdf['zip'])
# Determine means for numeric columns by zip
zip_mean_sales_df = zip_groupby.mean().sort_values(by=['salesvol'],
                                                   ascending=False)
# Calculate Zip ranks by average sales volume
zip_mean_sales_df['Zip Ave. Sales Rank'] = \
    zip_mean_sales_df['salesvol'].rank(ascending=False)

# Drop the unneeded 'SCORE' field
del zip_mean_sales_df['score']

zip_mean_sales_df.head()
```

zip	empnum	objectid	salesvol	Zip Ave. Sales Rank
15231	39.818182	4293.818182	1862.272727	1.0
15275	38.521739	3861.695652	1801.652174	2.0
15482	35.000000	950.000000	1637.000000	3.0
15084	32.605263	1093.789474	1524.842105	4.0
15205	30.424779	3716.203540	1422.796460	5.0

```
# Create a basic chart of the top five zip codes by sales volume
label = list(zip_mean_sales_df.head().index.values)
value = list(zip_mean_sales_df['salesvol'].head())

p = figure(x_range=label,
           title = 'Top Five Zip Codes by Mean Sales Volume',
           x_axis_label = 'Zip Code',
           y_axis_label = 'Average Sales volume in thousands of dollars')
p.xaxis.major_label_orientation = np.pi/4

p.vbar(x=label, top=value, width=0.5)

show(p)
```

```
# Load the geoenrichment module's country object for the US,
# and pre-cache some geometry
usa = Country.get("United States")
usa.subgeographies.states['Pennsylvania'].zip5['15275'].geometry
```

```
# Create a basic map showing the top five zip codes
top_five_zip_map = gis.map("Pittsburgh")
for zip5 in zip_mean_sales_df.head().index.values.tolist():
    top_five_zip_map.draw(usa.subgeographies.states['Pennsylvania']\
                          .zip5[str(zip5)].geometry)
restaurants_sdf.spatial.plot(map_widget=top_five_zip_map, alpha=0.1,
                             cmap='Greys', outline_color='Greys')
top_five_zip_map
```



esri Indonesia

Step 5: Perform spatial analysis

1. Untuk menentukan lokasi potensial maka kita akan melakukan klasterisasi berdasarkan keberadaan retail yang padat.
   - Tentukan lokasi dan nama dari data yang akan kita proses

```
# Set an output path name
output_path = "{0}//{1}_HDBSCAN".format(fgdb, "RestaurantClusters")
clusters_fc = arcpy.stats.DensityBasedClustering(fc, output_path,
                                                 "HDBSCAN", 5)
clusters_fc
```

```
<Result '/mnt/arcgis/me/pittsburgh_restaurants.gdb/RestaurantClusters_HDBSCAN'>
```

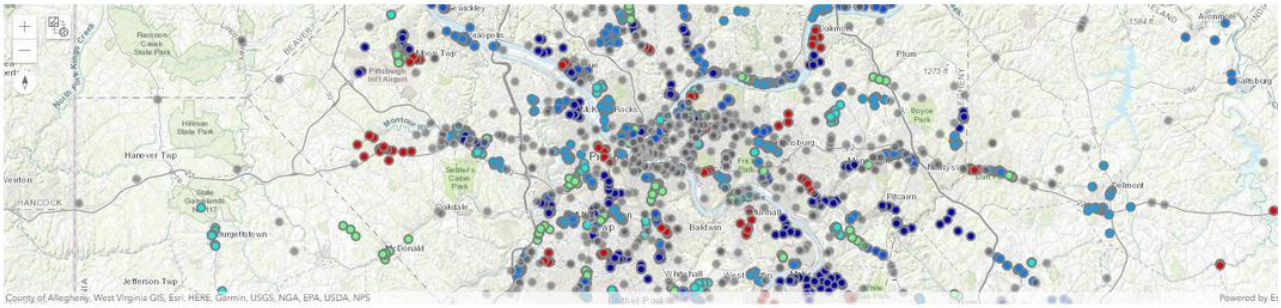
```
# Convert the clusters feature class from Density-based Clustering into
# a spatially-enabled dataframes (one for clusters, one for noise)
dbclustering_sdf = pd.DataFrame.spatial.from_featureclass(clusters_fc)
clusters_sdf = dbclustering_sdf.loc[dbclustering_sdf['CLUSTER_ID'] != -1]
noise_sdf = dbclustering_sdf.loc[dbclustering_sdf['CLUSTER_ID'] == -1]
clusters_sdf.head()
```

	OBJECTID	SOURCE_ID	CLUSTER_ID	PROB	OUTLIER	EXEMPLAR	STABILITY	COLOR_ID	SHAPE
3	4	4	56	0.173037	0.826963	0	0.0	8	{"x": -8904590.7839, "y": 4868224.023100004, "...
4	5	5	36	1.000000	0.000000	1	0.0	4	{"x": -8896241.8221, "y": 4844052.571800001, "...
5	6	6	36	1.000000	0.000000	1	0.0	4	{"x": -8895774.2802, "y": 4844531.075199999, "...
6	7	7	36	1.000000	0.000000	1	0.0	4	{"x": -8894527.5019, "y": 4845430.143799998, "...
7	8	8	36	0.790149	0.209851	0	0.0	4	{"x": -8893759.3974, "y": 4846532.338799998, "...

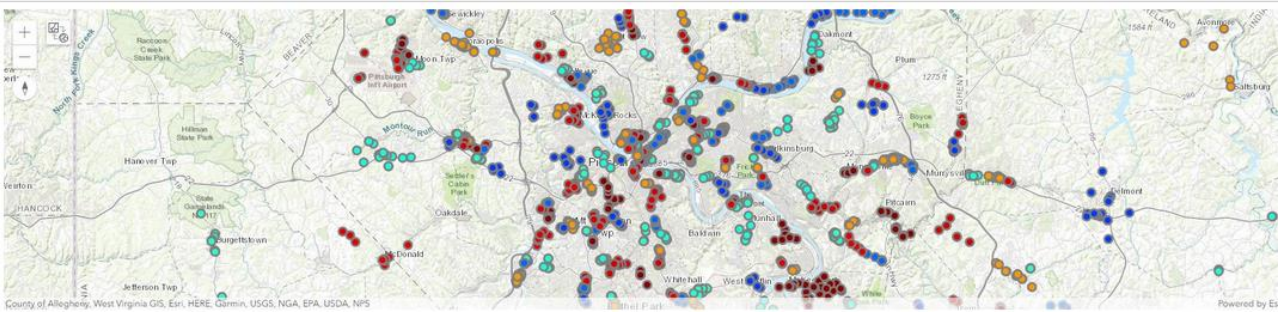
```
noise_sdf.head()
```

2. Visualisasi Data hasil klasterisasi

```
# Visualize both clusters and noise
results_map = gis.map("Pittsburgh")
clusters_sdf.spatial.plot(map_widget=results_map, renderer_type='u',
                          col="COLOR_ID")
noise_sdf.spatial.plot(map_widget=results_map, cmap='Greys', alpha=0.6,
                       outline_color='Greys')
results_map
```



```
# Visualize only clusters
onlyclusters_map = gis.map("Pittsburgh")
clusters_sdf.spatial.plot(map_widget=onlyclusters_map,
                          renderer_type='u', col="COLOR_ID")
onlyclusters_map
```



3. Pembuatan zonasi untuk tiap klister pada peta

```
# Run Directional Distribution using clusters as an input
clusters_fc = (
               clusters_sdf
               .copy()
               .spatial
               .to_featureclass(os.path.join(fgdb, "restaurant_clusters"))
              )
ellipses = arcpy.stats.DirectionalDistribution(clusters_fc,
                                               os.path.join(fgdb, "cluster_ellipses"),
                                               Ellipse_Size="2 standard deviations",
                                               Case_Field="CLUSTER_ID")[0]
ellipses_sdf = pd.DataFrame.spatial.from_featureclass(ellipses)
ellipses_sdf.head()
```

	OBJECTID	CenterX	CenterY	XStdDist	YStdDist	Rotation	CLUSTER_ID	SHAPE
0	1	-8.910530e+06	5.023269e+06	2151.586100	12525.834221	22.711797	1	{"rings": [[[-8912514.4278, 5024099.286899999]...
1	2	-8.825619e+06	4.985812e+06	11059.807891	19222.922180	23.821812	2	{"rings": [[[-8835736.6028, 4990278.838500001]...
2	3	-8.831966e+06	4.966738e+06	283.002577	4506.261331	143.274423	3	{"rings": [[[-8832192.7771, 4966569.239699997]...
3	4	-8.847666e+06	4.849466e+06	3419.763571	13281.725110	19.417593	4	{"rings": [[[-8850890.9565, 4850602.765900001]...
4	5	-8.875420e+06	5.007396e+06	3412.772598	12791.984258	14.330382	5	{"rings": [[[-8878726.6002, 5008240.671899997]...

```
print("Produced {0} ellipses.".format(ellipses_sdf.shape[0]))
```

```
Produced 346 ellipses.
```

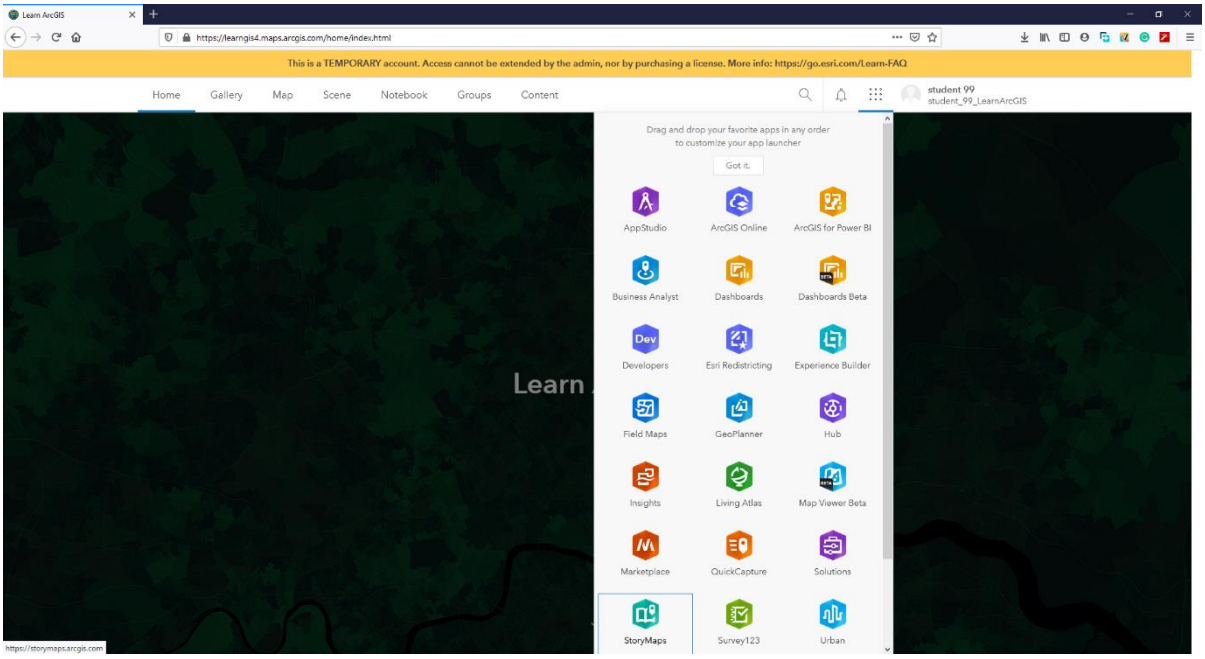
```
ellipses_map = gis.map("Pittsburgh")
clusters_sdf.spatial.plot(map_widget=ellipses_map,
                          renderer_type='u', col='CLUSTER_ID',
                          alpha=0.1, outline_color='Greys')
ellipses_sdf.spatial.plot(map_widget=ellipses_map, alpha=0.7,
                          outline_color='magma')
ellipses_map
```

4. Sampai tahap ini Analisa data spasial akan dilanjutkan disesi workshop. Atau Anda dapat mengakses materinya di https://data-scientist-learngis.hub.arcgis.com

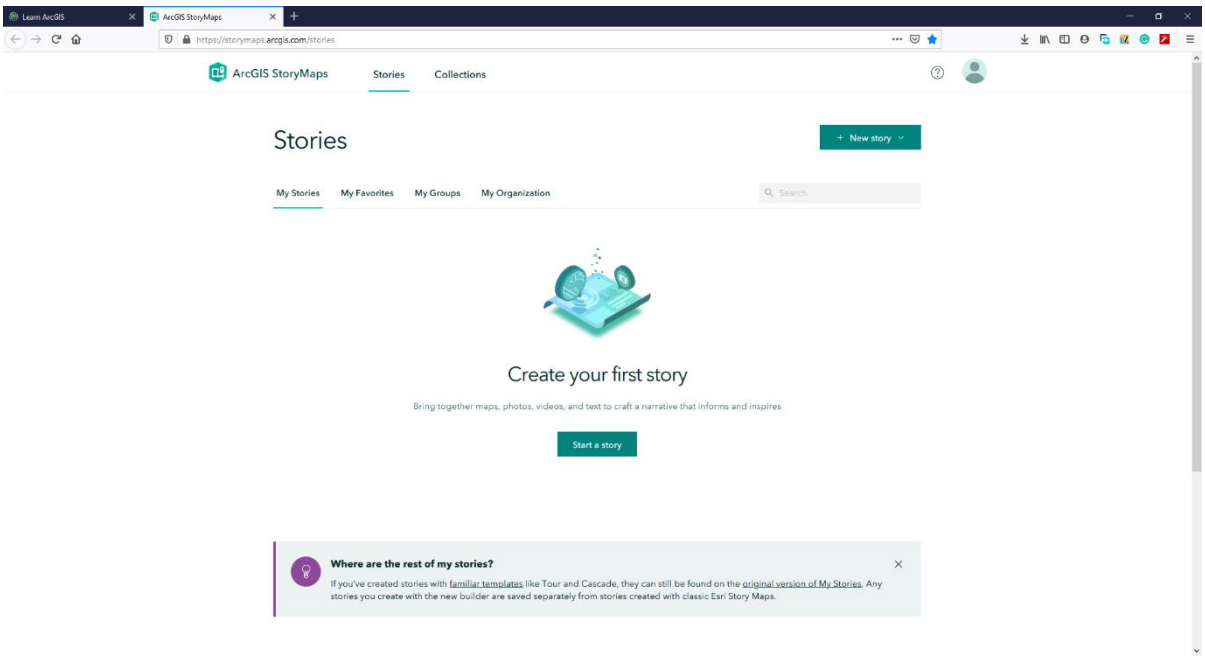
esri Indonesia

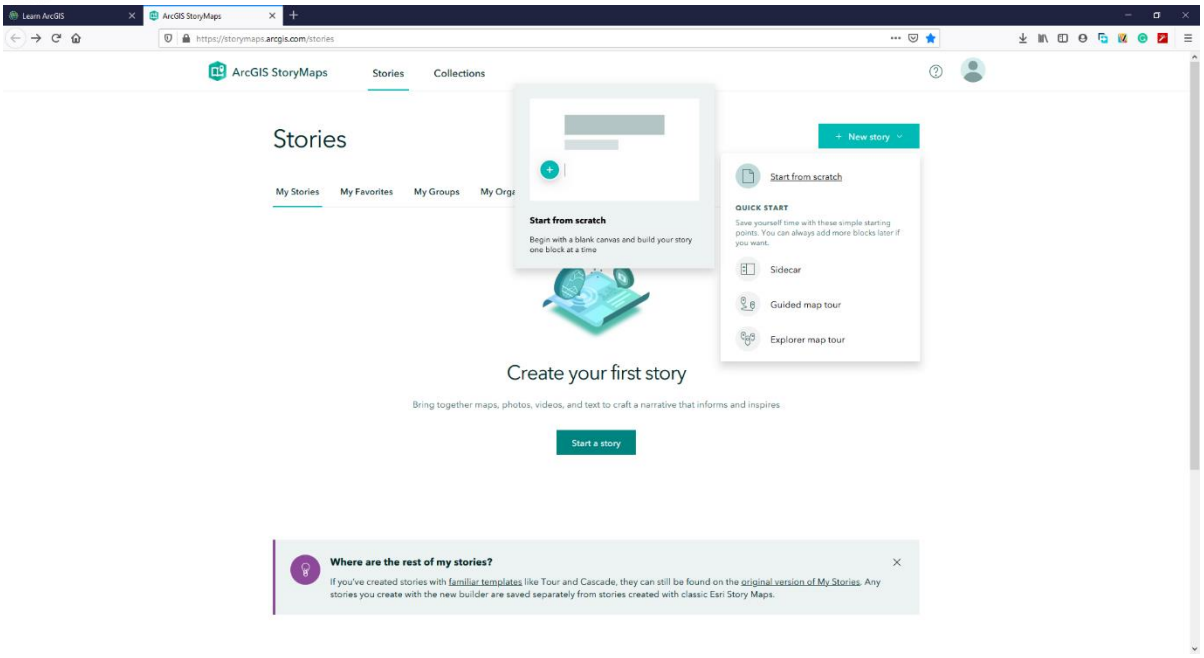
Step 6: Storytelling menggunakan ArcGIS StoryMap

1. Buka Kembali Home Page ArcGIS Online. Kemudian arahkan pada panel bagian kanan atas

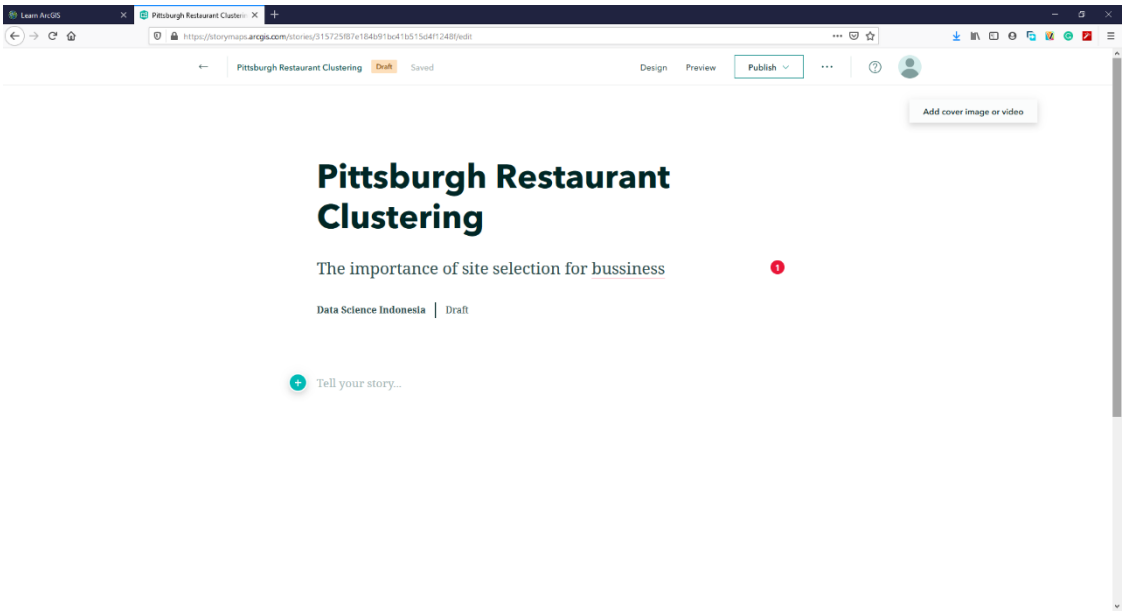


2. Klik button Start a story atau Anda bisa memilih button “New story”. Akan muncul beberapa pilihan dari membuat story dari *scratch* atau menggunakan *template*.

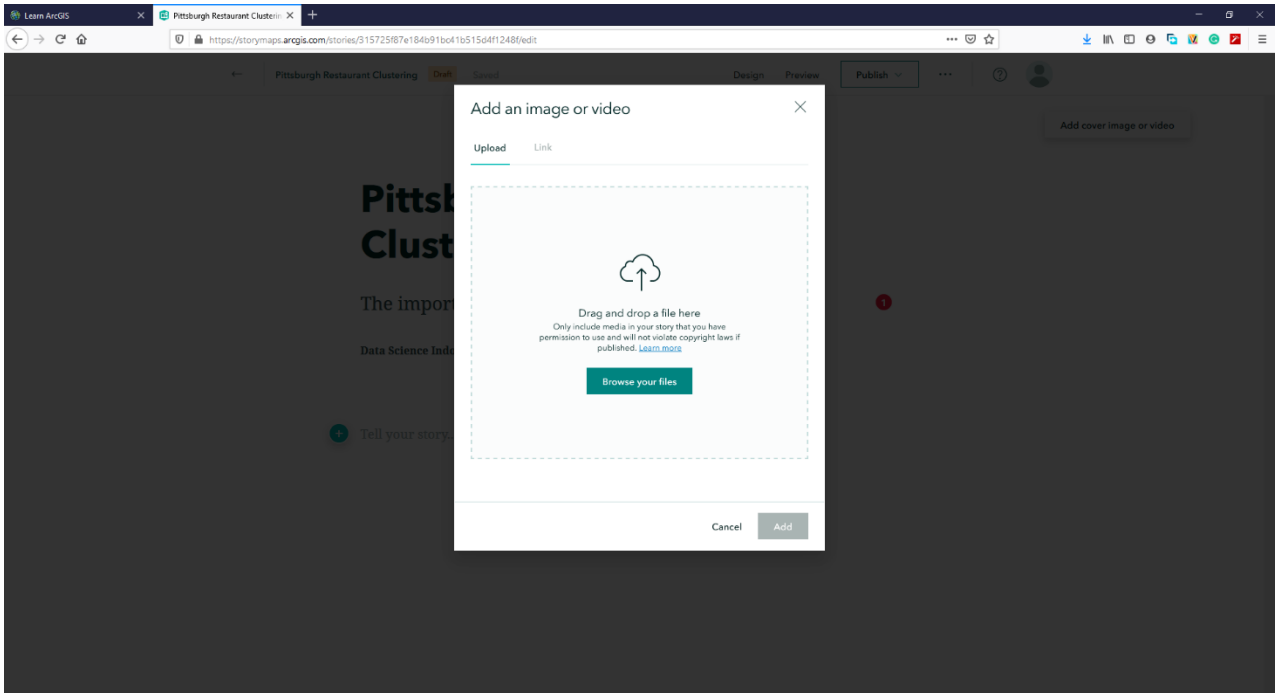




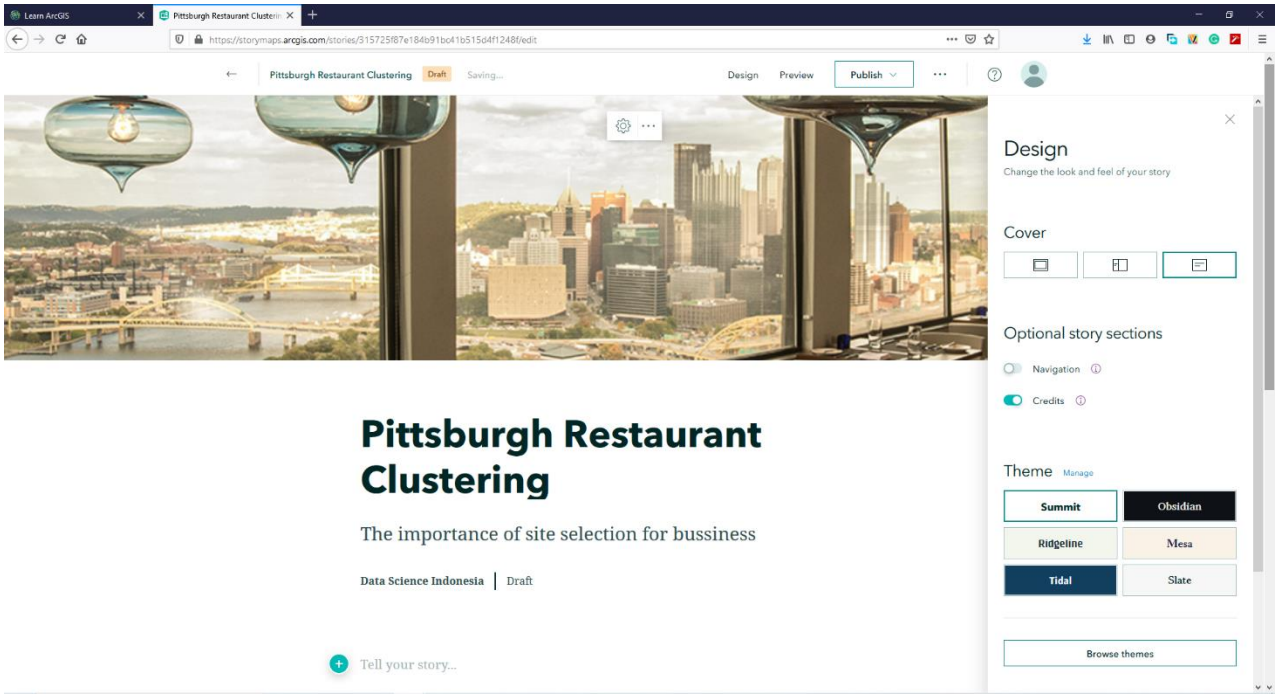
3. Mulailah menceritakan peta hasil visualisasi Anda dalam Story Map ini. Anda dapat mengawali nya dengan menentukan judul yang tepat dan background dengan latar belakang menarik. Contohnya seperti berikut



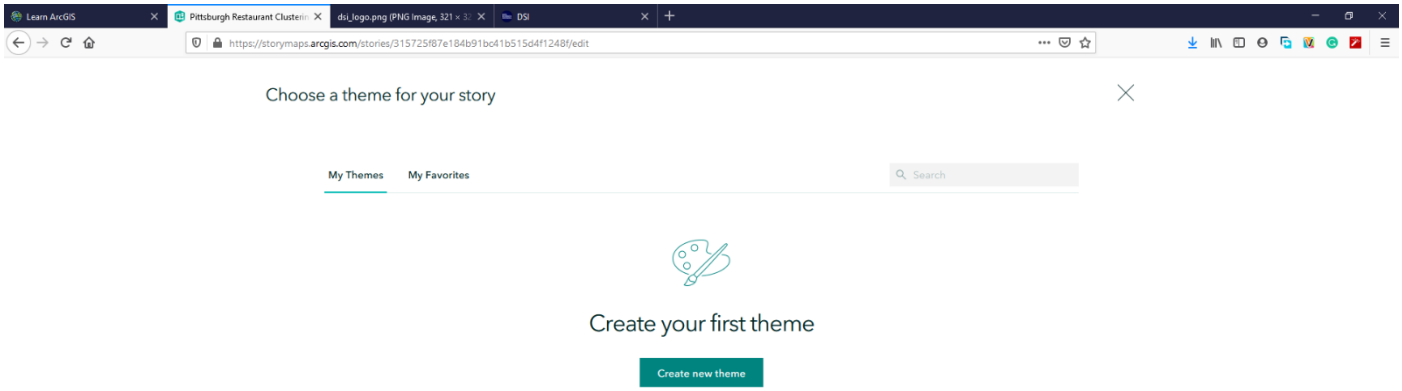
4. Tambahkan background dengan mengklik “Add cover images or video”. Kemudian browse dan cari background yang ingin Anda upload



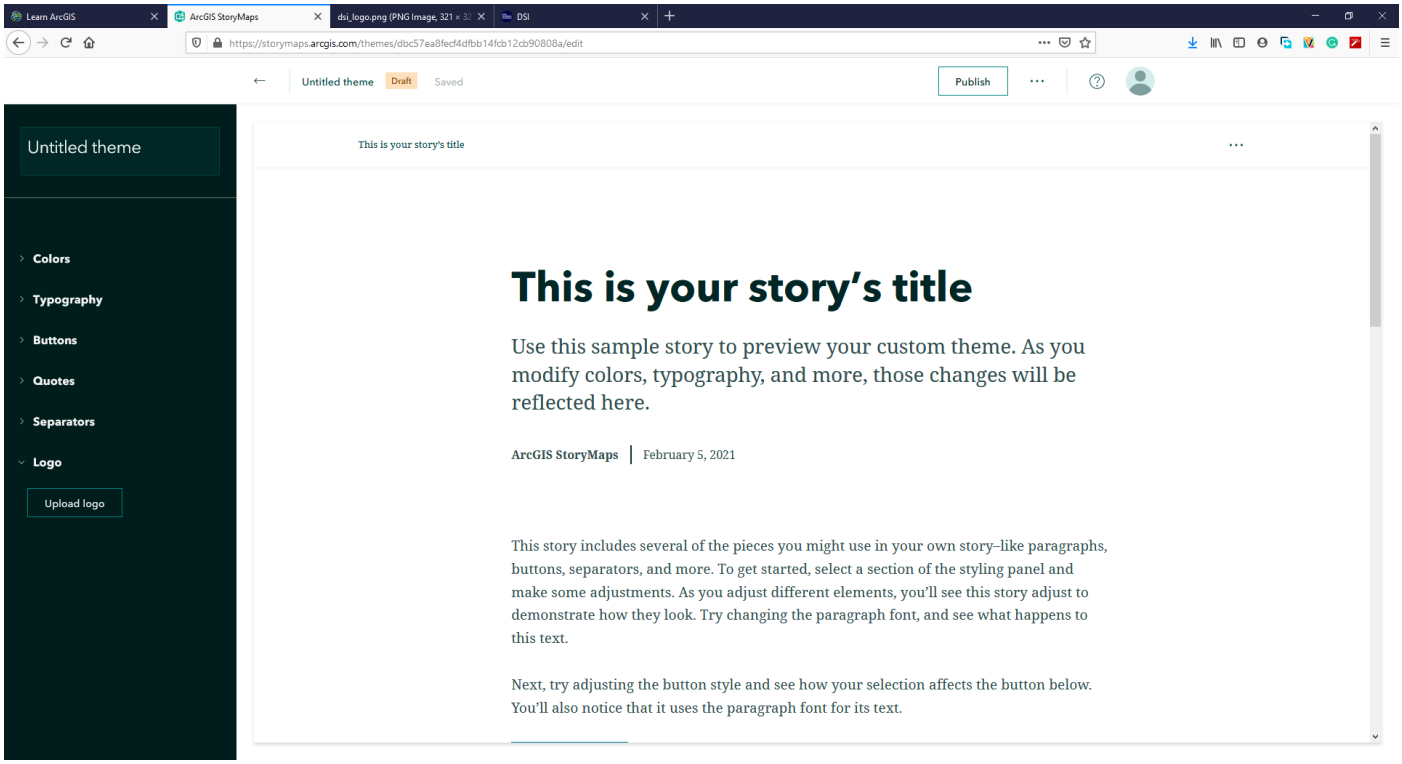
5. Anda juga bisa mengatur layout StoryMap melalui panel Design. Anda dapat mengatur warna backround hingga font.



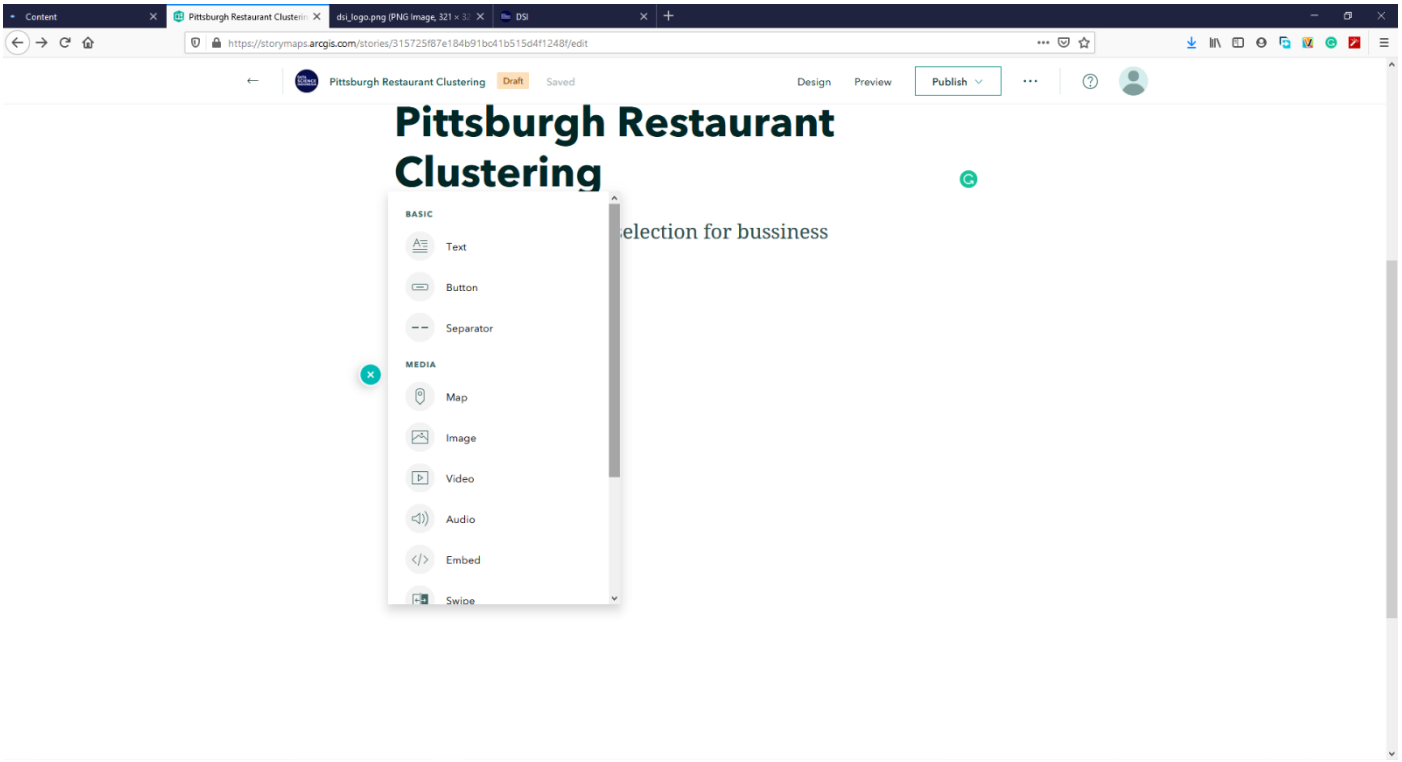
6. Anda juga bisa melakukan customisasi tata letak dan tema StoryMap dengan mengklik “Browse themes” -> Create new theme



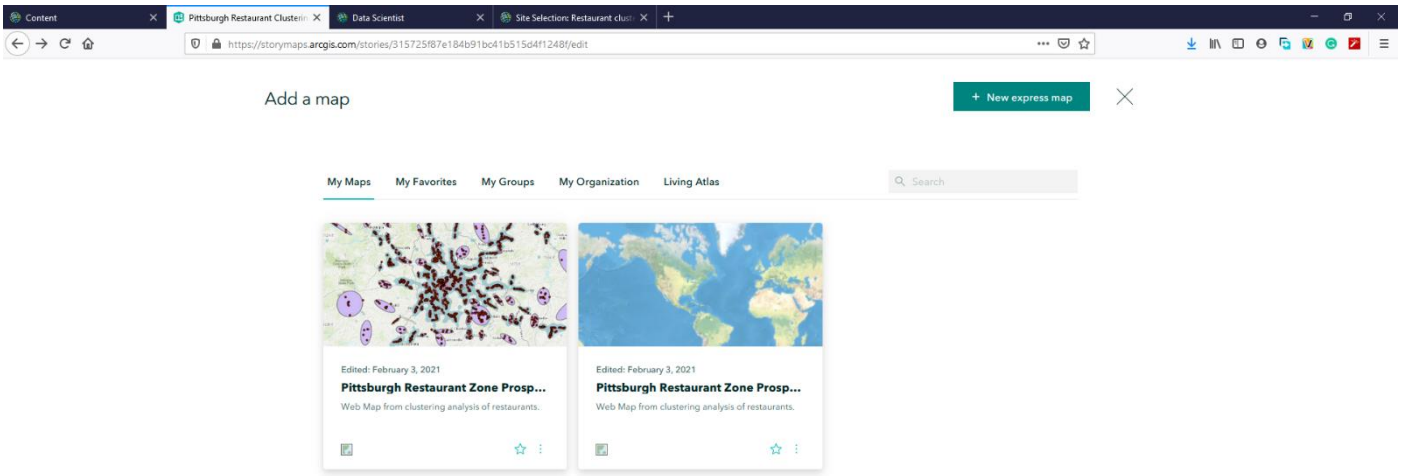
7. Anda dapat membuat template sesuai selera Anda pada bagian ini



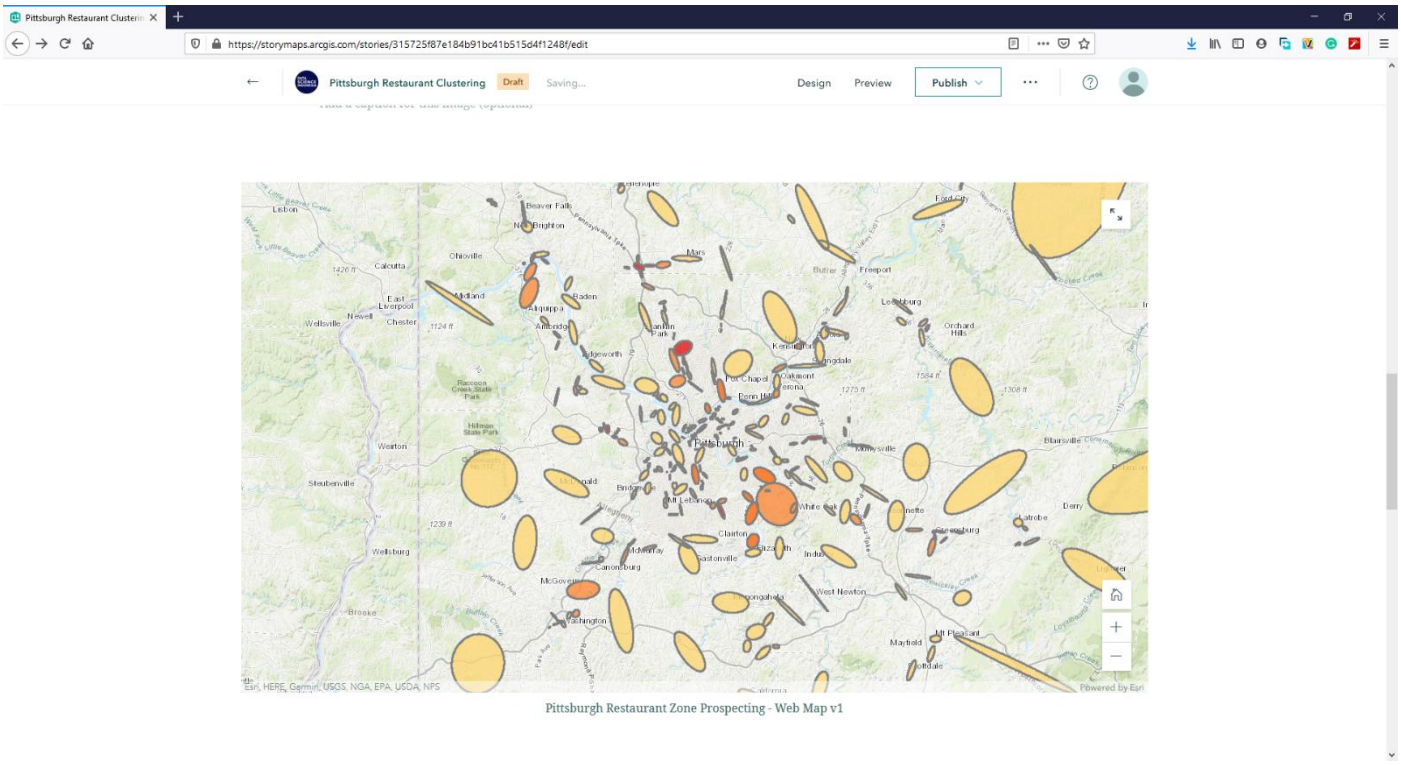
8. Kembali ke Laman utama StoryMap, Anda dapat menambahkan gambar, tulisan, peta, video hingga mengembed website ke StoryMap dengan mengklik tanda “+”



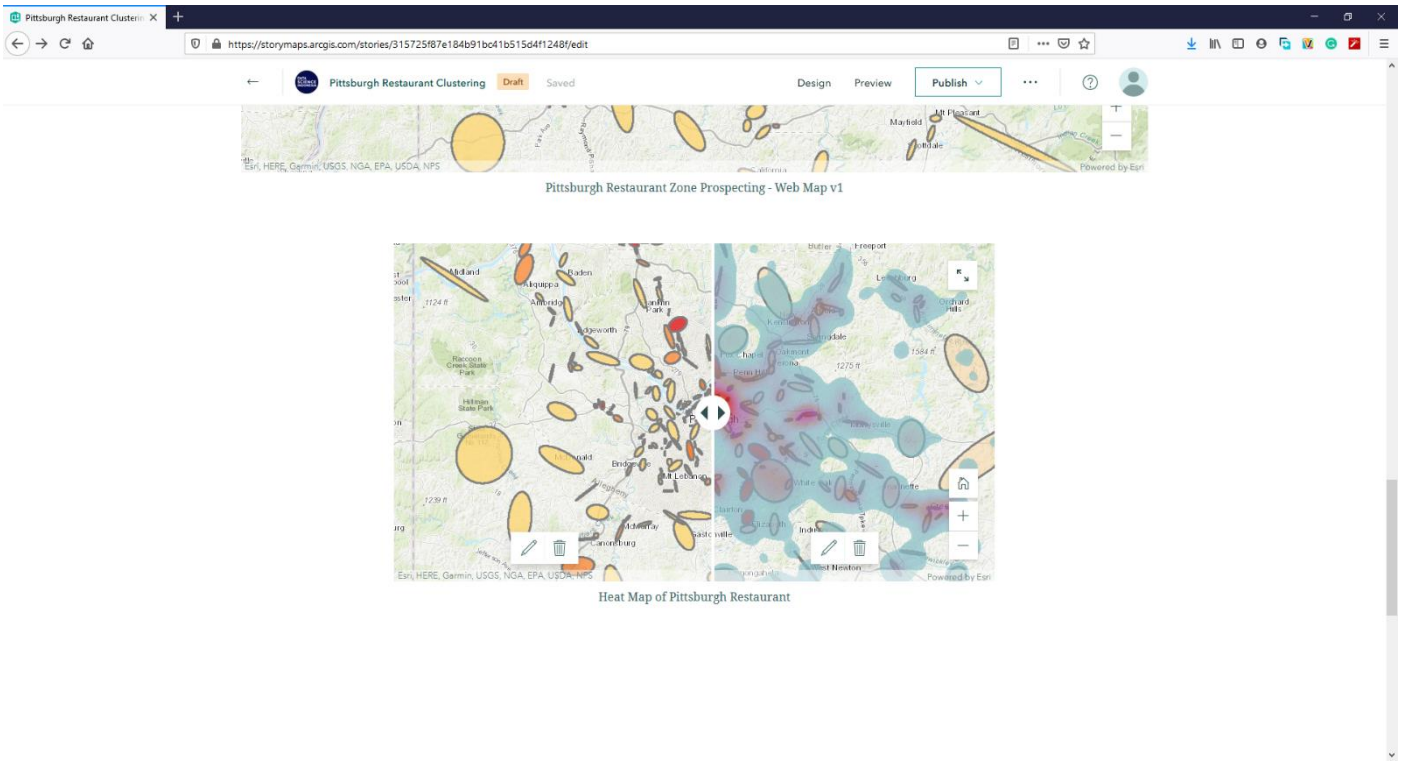
9. Apabila Anda ingin menambahkan Peta Anda sebelumnya, maka pilih Map dan akan tambil jendela seperti berikut



10. Anda dapat menampilkan peta dalam bentuk WebMap bahkan Swipeview seperti berikut



Pittsburgh Restaurant Zone Prospecting - Web Map v1



Pittsburgh Restaurant Zone Prospecting - Web Map v1

Heat Map of Pittsburgh Restaurant

Anda dapat melihat contoh StoryMap di link: https://arcg.is/1LeTjP