	<pre>import numpy as np import pandas as pd import matplotlib.pyplot as plt %matplotlib inline import seaborn as sns import re</pre>
In [3]:	<pre>import seaborn as sns import re import folium from folium import plugins 1. Data Ingestion data = pd.read_csv('Data Faskes BPJS 2019.csv', header=0)</pre>
In [4]:	MoLink Provinsi KotaKab Link TipeFaskes No KodeFaskes NamaFaskes LatLongFaskes AlamatFaskes TelpFaskes NamaFaskes LatLongFaskes AlamatFaskes Paskes AlamatFaskes TelpFaskes NamaFaskes LatLongFaskes AlamatFaskes Paskes AlamatFaskes Paskes AlamatFaskes Paskes Pask
	Darussalam Sakit RP1S di Kot la Sakit 2 0001R004 Medula n=4.488088 97 947781 Rebuil Balu 4840076\r\n
In [5]:	data.info() <class 'pandas.core.frame.dataframe'=""> RangeIndex: 29157 entries, 0 to 29156 Data columns (total 11 columns): # Column Non-Null Count Dtype</class>
	0NoLink29157 non-null int641Provinsi29157 non-null object2KotaKab29157 non-null object3Link29157 non-null object4TipeFaskes29157 non-null object5No29157 non-null object6KodeFaskes29157 non-null object7NamaFaskes29157 non-null object8LatLongFaskes29157 non-null object9AlamatFaskes29157 non-null object
	10 TelpFaskes 29157 non-null object dtypes: int64(1), object(10) memory usage: 2.4+ MB 2. Data Cleaning Bersihkan nama dan spasi
In [7]:	<pre>def cleanNama(row): str = row.replace('Kode Faskes dan Alamat Rumah Sakit BPJS di','') return str def removeSpace(row): str = ' '.join(row.split()) return str data['KotaKab'] = data['KotaKab'].apply(removeSpace).apply(cleanNama)</pre>
Out[7]:	data['NamaFaskes'] = data['NamaFaskes'].apply(removeSpace) data['TelpFaskes'] = data['TelpFaskes'].apply(removeSpace) data.drop(['Link','NoLink'], axis=1, inplace=True) data.head() Provinsi KotaKab TipeFaskes No KodeFaskes NamaFaskes LatLongFaskes AlamatFaskes TelpFaskes 0 Nanggroe Aceh Darussalam Kota Langsa Rumah Sakit 1 0001R001 RSU Cut Nyak Dhien http://maps.google.co.id/?q=4.488058,97.947963 Jl. Tm Bahrum No. 1 Langsa 0641-0621039
	1 Nanggroe Aceh Darussalam Kota Langsa Rumah Sakit 2 0001R004 RSU Cut Meutia Langsa http://maps.google.co.id/?q=4.488088,97.947781 Jl.Garuda Kebun Baru Langsa 0641-4840076 2 Nanggroe Aceh Darussalam Kota Langsa Rumah Sakit 3 0105R001 RSUD Langsa http://maps.google.co.id/?q=4.472208,97.975533 Jln.A.Yani No. 1 Langsa 0641-22051 3 Nanggroe Aceh Darussalam Kota Langsa Rumah Sakit 4 0105R013 RSU Ummi http://maps.google.co.id/?q=4.470376,97.991915 Jln. Prof. A. Madjid Ibrahim 0641-22886 4 Nanggroe Aceh Darussalam Kota Langsa Puskesmas 1 00010001 Langsa Lama http://maps.google.co.id/?q=4.478172,97.949988 Gampong Meurandeh Dayah 0641-064121218 Bersihkan latitude dan longitude
In [8]:	def searchLatLong(row): str = re.search('(-?([0-9]{1} [0-9][1-8]{1,2}).[0-9]{1,6},(-?(1[0-8]{1,2} 9[1-9]{1})).[1-9]{1,6})))', row) if str: return str.group() return np. NaN data['LatLongFaskes'] = data['LatLongFaskes'].apply(searchLatLong) Lat = [] for row in data['LatLongFaskes']; try: latitude = float(row.split(', ')[0]) longitude = float(row.split(', ')[0]) longitude = float(row.split(', ')[1]) if (-90.0 <= latitude <= 90.0); lat.append(latitude) else: lat.append(np.NaN) if (-180 <= longitude <= 180): lon.append(longitude) else: lon.append(np.NaN) except: lat.append(np.NaN) data['Latitude'] = lat data('Longitude'] = lon data dropt['latlongFaskes'], axis=1, inplace=True) data.head() Provinsi KotaKab TipeFaskes No KodeFaskes NamaFaskes AlamatFaskes TelpFaskes Latitude Longitude
	0 Nanggroe Aceh Darussalam Kota Langsa Rumah Sakit 1 0001R001 RSU Cut Nyak Dhien Jl. Tm Bahrum No. 1 Langsa 0641-0621039 4.488058 97.947963 1 Nanggroe Aceh Darussalam Kota Langsa Rumah Sakit 2 0001R004 RSU Cut Meutia Langsa Jl. Garuda Kebun Baru Langsa 0641-4840076 4.488088 97.947781 2 Nanggroe Aceh Darussalam Kota Langsa Rumah Sakit 3 0105R001 RSUD Langsa Jln. A.Yani No. 1 Langsa 0641-22051 4.472208 97.975533 3 Nanggroe Aceh Darussalam Kota Langsa Rumah Sakit 4 0105R013 RSU Ummi Jln. Prof. A. Madjid Ibrahim 0641-22886 4.470376 97.991915 4 Nanggroe Aceh Darussalam Kota Langsa Puskesmas 1 00010001 Langsa Lama Gampong Meurandeh Dayah 0641-064121218 4.478172 97.949988
	<pre>3. Data Analysis 1. Provinsi dengan Fasilitas Kesehatan ber-BPJS provinsi = data['Provinsi'].unique().tolist() print(provinsi) print('\nTotal provinsi: ', len(provinsi)) ['Nanggroe Aceh Darussalam', 'Sumatera Utara', 'Sumatera Barat', 'Riau', 'Jambi', 'Sumatera Selatan', 'Bengkulu', 'Lampung', 'Kep. Bangka Belitung', 'Dki Jaka</pre>
	rta', 'Jawa Barat', 'Jawa Tengah', 'D I Yogyakarta', 'Jawa Timur', 'Banten', 'Bali', 'Nusa Tenggara Barat', 'Nusa Tenggara Timur', 'Kalimantan Barat', 'Kalimantan Tengah', 'Kalimantan Selatan', 'Kalimantan Timur', 'Sulawesi Utara', 'Sulawesi Tengah', 'Sulawesi Selatan', 'Sulawesi Tenggara', 'Gorontalo', 'Maluku', 'Maluku Utara', 'Papua', 'Papua Barat', 'Sulawesi Barat', 'Kepulauan Riau', 'Kalimantan Utara'] Total provinsi: 34 2. Kota/Kabupaten dengan Fasilitas Kesehatan ber-BPJS kabkota = data['KotaKab'].unique().tolist() print(kabkota)
	Pieter Michael Receit Aden Aden Selaton Kob. Acen Selaton
	Total kota: 512 Geographically available and unavailable data total_data_count = data.shape[0] geographically_available_data_count = data.dropna().shape[0] geographically_unavailable_data_count = data[data['Latitude'].isnull() data['Longitude'].isnull()].shape[0]
	print('\nTotal data fasilitas: ', total_data_count) print('\nTotal fasilitas yang dapat ditampilkan di map: ', geographically_available_data_count) print('\nTotal fasilitas yang tidak dapat ditampikan di map: ', geographically_unavailable_data_count) Total data fasilitas: 29157 Total fasilitas yang dapat ditampilkan di map: 16499 Total fasilitas yang tidak dapat ditampikan di map: 12658
,	4. Data Visualization Visualisasi geografis 16499 fasilitas kesehatan ber-BPJS Visualisasi data dengan data geografis dengan folium FastMarkCluster data_map = data.dropna()
	data_map = data.dropha() <class 'pandas.core.frame.dataframe'=""> Int64Index: 16499 entries, 0 to 29149 Data columns (total 10 columns): # Column Non-Null Count Dtype </class>
	TipeFaskes 16499 non-null object None 16499 non-null object NamaFaskes 16499 non-null object AlamatFaskes 16499 non-null object AlamatFaskes 16499 non-null object TelpFaskes 16499 non-null object Latitude 16499 non-null float64 Longitude 16499 non-null float64 dtypes: float64(2), object(8)
n [13]:	<pre>data_map = data.dropna() rome_lat, rome_lng = -6.200000, 106.816666 # init the folium map object my_map = folium.Map(location=[rome_lat,rome_lng],zoom_start=5) # add all the point from the file to the map object using FastMarkCluster my_map.add_child(plugins.FastMarkerCluster(data_map[['Latitude','Longitude']].values.tolist()))</pre>
	Col Burul Barida Ach Ach Self Ach 1910
n [14]: ut[14]:	<pre>data_kotakab = data['KotaKab'].value_counts().rename_axis('KotaKab').reset_index(name='Jumlah') data_kotakab.head() KotaKab Jumlah Kota Surabaya 364</pre>
n [15]:	1 Kota Semarang 305 2 Kab. Bogor 298 3 Kota Makassar 292 4 Kota Palembang 288 # Mengambil latitude dan longitude data per kota/kabupaten
	<pre>kabkota = [] prov = [] latk = [] lonk = [] data_n = data.dropna() #hapus kota/kab tanpa latitude/longtitude agar data dapat dibuat map for prv, kk, ltk, lnk in zip(data_n['Provinsi'], data_n['KotaKab'], data_n['Latitude'], data_n['Longitude']): if (not(kk in kabkota) and not(ltk == np.NaN) and not(lnk == np.NaN)): prov.append(prv)</pre>
	<pre>kabkota.append(kk) latk.append(ltk) lonk.append(lnk) print(len(kabkota)) data_map = pd.DataFrame(list(zip(prov, kabkota, latk, lonk)), columns=['Provinsi', 'KotaKab', 'Latitude', 'Langitude']) data_map.head()</pre>
	Provinsi KotaKab Latitude Langitude Nanggroe Aceh Darussalam Kota Langsa 4.488058 97.947963 Nanggroe Aceh Darussalam Kab. Gayo Lues 4.059445 97.463333 Nanggroe Aceh Darussalam Kab. Aceh Selatan 3.111916 97.368642
n [16]:	3 Nanggroe Aceh Darussalam Kab. Aceh Tenggara 3.496266 97.800000
ut[16]:	3 Nanggroe Aceh Darussalam Kab. Aceh Tenggara 3.496266 97.800000 4 Nanggroe Aceh Darussalam Kab. Aceh Timur 4.961196 97.762670 merged = pd.merge(data_map, data_kotakab, on='KotaKab') merged.count() Provinsi 460 KotaKab 460
	3 Nanggroe Aceh Darussalam Kab. Aceh Tenggara 3.496266 97.800000 4 Nanggroe Aceh Darussalam Kab. Aceh Timur 4.961196 97.762670 merged = pd.merge(data_map, data_kotakab, on='KotaKab') merged.count() Provinsi 460
	<pre>3 Nanggroe Aceh Darussalam Kab. Aceh Tenggara 3.496266 97.800000 4 Nanggroe Aceh Darussalam Kab. Aceh Timur 4.961196 97.762670 merged = pd.merge(data_map, data_kotakab, on='KotaKab') merged.count() Provinsi</pre>
n [17]:	<pre>3 Nanggroe Aceh Darussalam Kab. Aceh Timur 4.961196 97.800000 4 Nanggroe Aceh Darussalam Kab. Aceh Timur 4.961196 97.762670 merged = pd.merge(data_map, data_kotakab, on='KotaKab') merged.count() Provinsi 460 KotaKab 460 Langitude 460 Langitude 460 Jumlah 460 dtype: int64 # Create map bpjs_map = folium.Map(location=[10, -20], zoom_start=2.3, tiles='Stamen Toner') for lat, lon, prov, kotakab, jml in zip(merged['Latitude'], merged['Langitude'], merged['Provinsi'], merged['KotaKab'], merged['Jumlah']):</pre>
n [17]:	**S Nanggroe Aceh Darussalam Kab. Aceh Timur 4,961106 97.762670 merged = pd.merge(data_map, data_kotakab, on='Kotakab') merged = pd.merge(data_map, data_kotakab, on='Kotakab') merged = count() **Frowing i
ut[17]:	3 Newton Nazaram Kah Acen Tengaya Seleptis Officials 4 Newton Nazaram Tengaya Seleptis Official
n [17]:	3 Restance Aced Deceasion Not Aced Tensus 3-84505 97.000000 4 Restaurce Aced Deceasion Not Aced Tensus 4301183 97.702710 merged = pd.morgo(stat map, cata Acetasch, com*Noracch*) merged = pd.morgo(stat map, cata Acetasch, com*Noracch*) merged = pd.morgo(stat map, cata Acetasch, com*Noracch*) PPRIVIDENT Aced Acetasch 400 Linklink 400 disput 11044 **Comate aced Acetasch 400 Linklink 400 **Data Acetasch 400 Linklink 400
n [17]:	All Registration Receives Test And Transport Calculated (Series Statement Tester) ### Provided And Calculated And Calculated (Series Statement Tester) #### Provided And Calculated And Calculated (Series Statement Tester) ##### Provided And Calculated And Calculated (Series Statement Tester) ###################################
it[17]:	Security of Control Security Control Sec
n [17]: n [18]:	Statistics Process Statistics Process Statistics Statistic
n [17]: n [18]: n [21]:	The proper of th
[17]: It [18]:	# A MERCHAN AND AND AND AND AND AND AND AND AND A
[17]: It [17]: It [18]:	Figure 1 and the content of the cont
t[17]: it[17]: it[17]: it[21]:	Security of the property of th
[17]: [17]: [18]: [21]:	The second of th
[17]: it [17]: it [17]: it [21]: it [22]: it [22]:	A CANCELLA CONTROLLA CONTR
n [17]: nt [17]: nt [17]: nt [18]: nt [21]: nt [22]: nt [23]: nt [24]:	A Company of the Comp
n [17]: ut [17]: ut [17]: ut [21]: ut [22]: ut [23]:	Secretary of the control of the cont
n [17]: ut [17]: ut [17]: ut [21]: ut [22]: ut [23]:	Sequence of the control of the contr
n [17]: ut [17]: ut [17]: ut [21]: ut [22]: ut [23]:	The property of the control of the c
n [17]: ut [17]: ut [17]: ut [21]: ut [22]:	
n [17]: ut [17]: ut [17]: ut [21]: ut [22]: ut [23]:	The property of the control of the c
n [17]: ut[17]: ut[17]: ut[21]:	The property of the control of the c
n [17]: ut [17]: ut [17]: ut [21]: ut [22]: ut [23]:	The part of the control of the contr
n [17]: nt [17]: nt [17]: nt [18]: nt [21]: nt [22]: nt [23]: nt [24]:	The part of the control of the contr