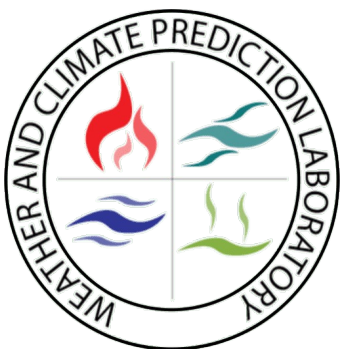


Cara Cepat Belajar Dasar – Dasar Pemetaan Menggunakan Cartopy

Sandy Hardian Susanto Herho



Pendahuluan

Cartopy merupakan pustaka Python yang digunakan untuk proses pembuatan peta dan ditujukan untuk menggantikan pustaka basemap yang rencananya tidak akan diteruskan pada akhir tahun 2020 ini. Seperti pada basemap, Cartopy juga dijalankan di atas pustaka matplotlib. Cartopy bergantung pada beberapa pustaka Python lainnya seperti geos dan shapely, oleh karena itu instalasi menggunakan pip akan sangat sulit untuk dilakukan pemula. Maka dari itu, pada tutorial ini saya menyarankan kepada para pengguna untuk melakukan instalasi dengan menggunakan [Miniconda 3](#). Sesudah melakukan instalasi Miniconda 3, jalankan perintah sebagai berikut di Terminal pengguna masing - masing:

```
conda install -c conda-forge cartopy
```

Selain itu, pembaca juga diharapkan untuk melakukan instalasi pustaka pandas yang digunakan untuk pembacaan data csv pada bagian akhir tutorial singkat ini:

```
conda install -c anaconda pandas
```

Pada tutorial ini, penyusun juga akan menggunakan Jupyter Notebook sebagai lingkungan pengembangannya. Diharapkan pengguna juga menggunakan lingkungan pengembangan yang sama, oleh karena itu, jalankan perintah sebagai berikut di Terminal kalian masing - masing:

```
conda install -c anaconda jupyter
```

Untuk memulai sesi Jupyter Notebook, jalankan perintah:

```
jupyter notebook
```

Maka secara otomatis peramban web bawaan kalian akan membuka sesi Jupyter Notebook.

Membuat peta pertama

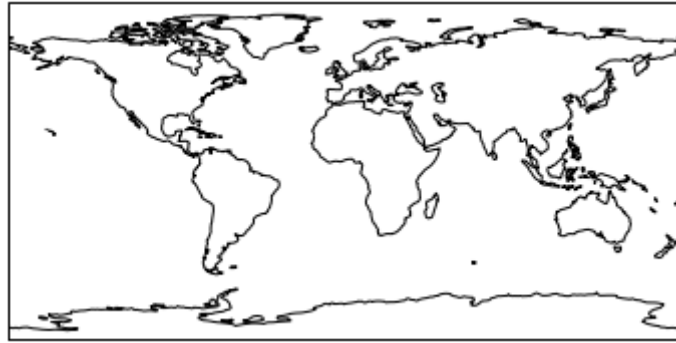
Hal pertama yang harus kita lakukan adalah mengimpor beberapa pustaka sebagai berikut:

```
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
%matplotlib inline
```

Pada baris pertama kita mengimpor modul pyplot di pustaka matplotlib (karena Cartopy dibangun di atas matplotlib) dan pada baris kedua kita mengimpor modul crs (*coordinate reference system*) pada pustaka Cartopy untuk melakukan pemetaan. Pada baris terakhir sendiri kita menjalankan perintah *magic* untuk menampilkan grafik matplotlib di Notebook tanpa perlu menjalankan perintah

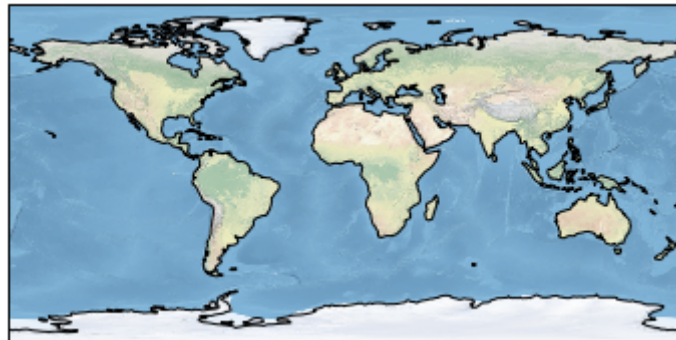
`plt.show()` setiap kali hendak menampilkan grafik.

```
ax = plt.axes(projection=ccrs.PlateCarree()) # mendefinisikan proyeksi
ax.coastlines(); # menampilkan garis pantai
```



Kemudian kita menggunakan gaya visualisasi berorientasi objek pada matplotlib dengan mendefinisikan sumbu dengan proyeksi dua dimensi (*Plate Carrée*) dan kemudian menampilkan garis pantai pada peta tersebut. Secara *default* latar belakang yang ditampilkan berwarna putih. Untuk menampilkan latar belakang yang berwarna, kita dapat menjalankan metode `stock_img` pada objek `ax`.

```
ax = plt.axes(projection=ccrs.PlateCarree()) # mendefinisikan proyeksi
ax.coastlines(); # menampilkan garis pantai
ax.stock_img(); # menampilkan peta berwarna
```

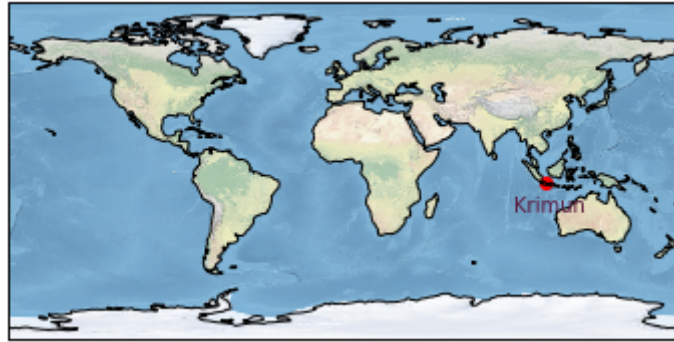


Kemudian kita akan mencoba menampilkan lokasi penyusun bermukim saat ini di Desa Krimun, Losarang, Indramayu, Jawa Barat (108.157691, -6.388166) dengan menggunakan *scatterplot*:

```
ax = plt.axes(projection=ccrs.PlateCarree()) # mendefinisikan proyeksi
ax.coastlines(); # menampilkan garis pantai
ax.stock_img(); # menampilkan peta berwarna

# Koordinat Desa Krimun, Losarang, Indramayu, Jawa Barat
bujur = 108.157691
lintang = -6.388166

plt.scatter(bujur, lintang, color='red'); # menampilkan titik berwarna merah
# menampilkan teks Krimun dgn jarak beberapa inci dari titik
plt.text(bujur - 17, lintang - 15, 'Krimun', color='#5c1a41');
```

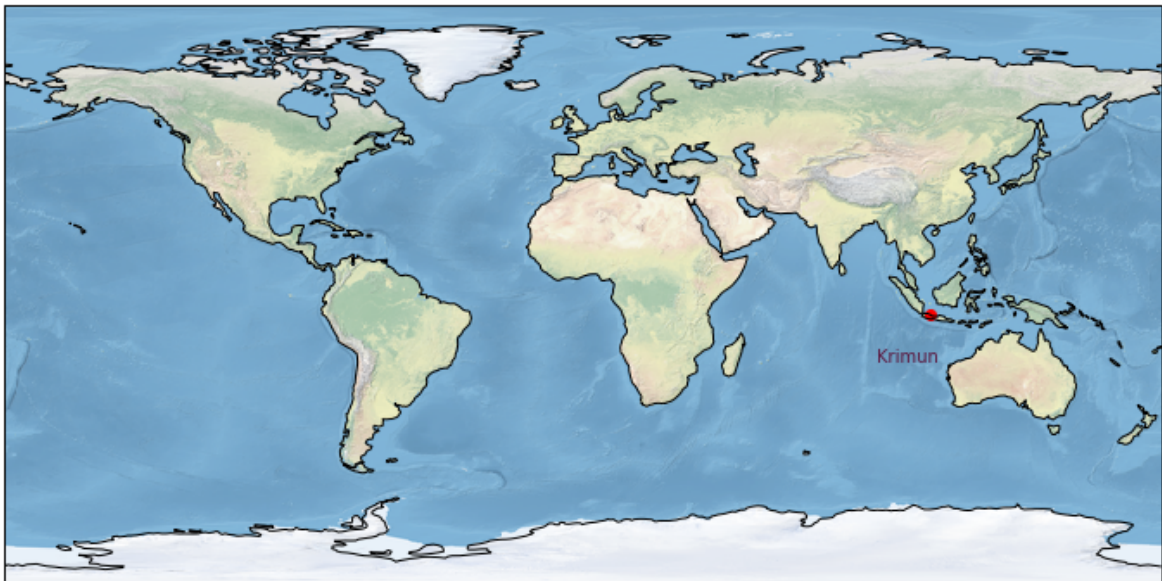


Untuk mengatur ukuran peta, kita dapat menggunakan metode `figure()` di modul `pyplot` di bagian awal *script* dengan menambahkan argumen `figsize` dalam bentuk *tuple* (lebar, tinggi) dalam satuan inci dalam kasus ini ukuran peta adalah 12×10 inci:

```
plt.figure(figsize=(12,10))
ax = plt.axes(projection=ccrs.PlateCarree()) # mendefinisikan proyeksi
ax.coastlines(); # menampilkan garis pantai
ax.stock_img(); # menampilkan peta berwarna

# Koordinat Desa Krimun, Losarang, Indramayu, Jawa Barat
bujur = 108.157691
lintang = -6.388166

plt.scatter(bujur, lintang, color='red'); # menampilkan titik berwarna merah
# menampilkan teks Krimun dgn jarak beberapa inci dari titik
plt.text(bujur - 17, lintang - 15, 'Krimun', color='#5c1a41');
```



Sekilas tentang proyeksi peta

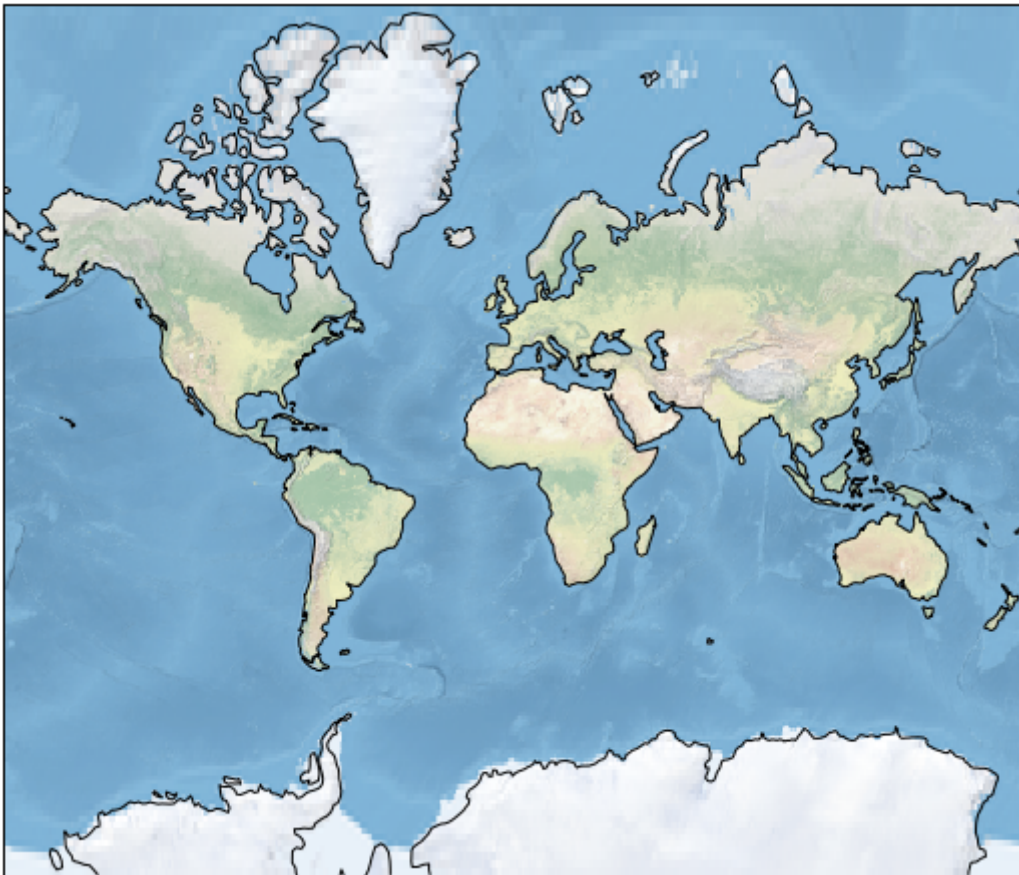
Ketika menampilkan permukaan bumi yang bersifat 3D ke lembar peta 2D, maka pasti terjadi distorsi. Oleh karena itu, untuk mengurangi distorsi tersebut terdapat berbagai macam proyeksi yang dapat digunakan. Suatu peta yang ideal harus memenuhi ketiga syarat sebagai berikut:

- *Conformal*: artinya adalah suatu proyeksi peta yang ideal harus mempertahankan fitur asli dari rupa bumi, utamanya pada kemiringan sudutnya (namun berbeda pada fitur luasan dan panjang rupa bumi).

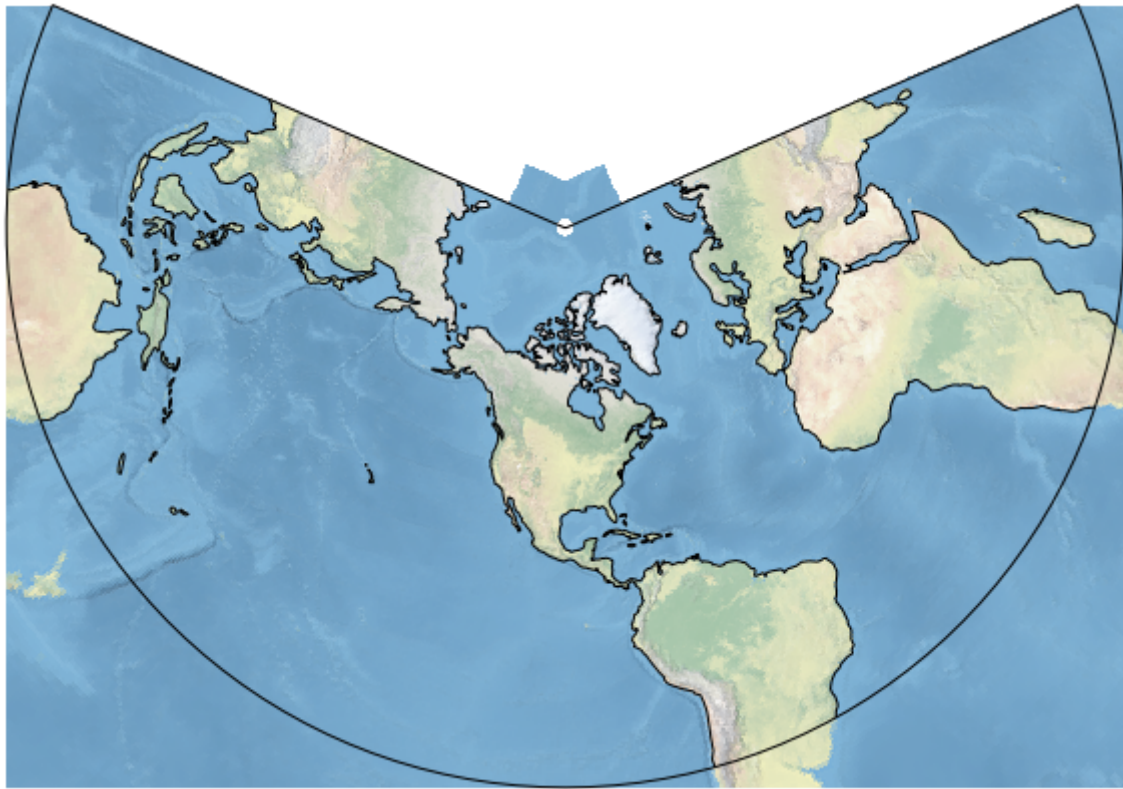
- *Equal area* : Proyeksi yang bersifat *equal area* merupakan proyeksi yang merujuk pada luasan yang sama antara di peta dengan rupa bumi aslinya (namun terdapat distorsi pada bentuk, sudut, dan/atau skalanya).
- *Equidistant*: Proyeksi yang bersifat *equidistant* adalah proyeksi yang mempertahankan kesamaan jarak dari bagian tengah peta, dengan kata lain panjang jarak antar meridian harus seragam.

Namun, tidak ada satupun proyeksi peta yang dapat memenuhi ketiga syarat ideal tersebut.

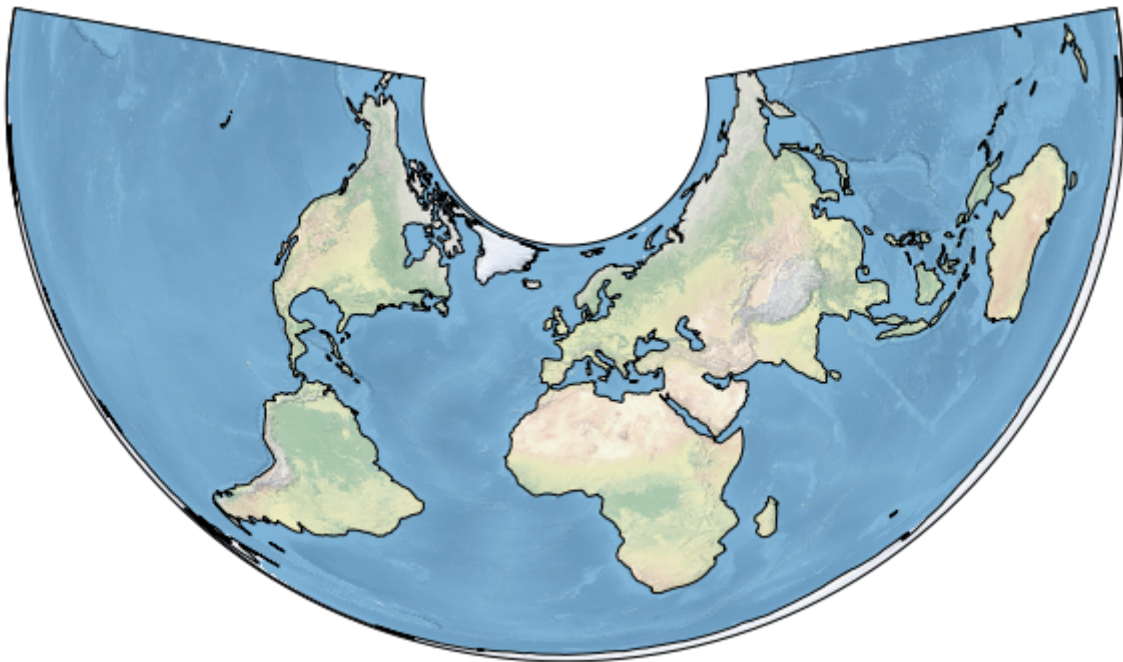
```
# Contoh proyeksi dengan sifat conformal, yakni Mercator
plt.figure(figsize=(10,8))
ax = plt.axes(projection=ccrs.Mercator())
ax.coastlines();
ax.stock_img();
```



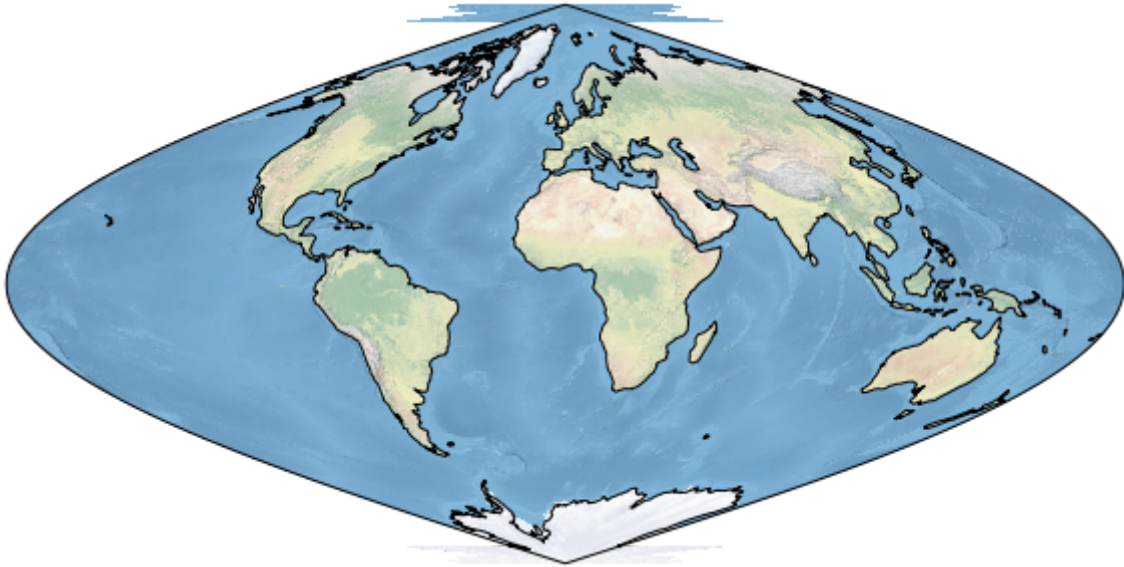
```
# Contoh proyeksi dengan sifat conformal, yakni LambertConformal
plt.figure(figsize=(10,8))
ax = plt.axes(projection=ccrs.LambertConformal())
ax.coastlines();
ax.stock_img();
```

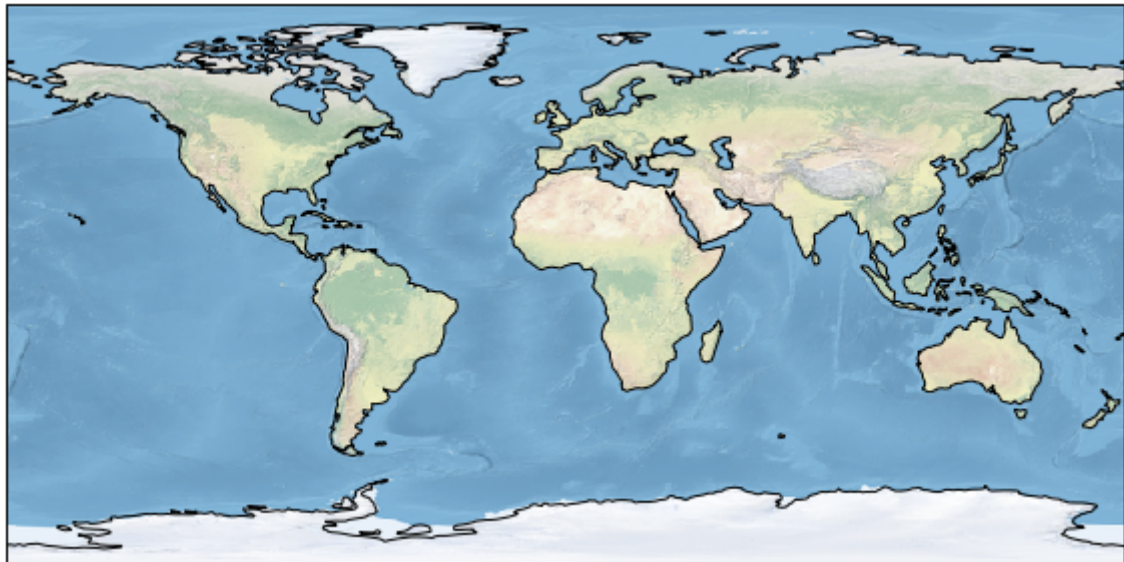
```
# Contoh proyeksi dengan sifat equal area, yakni AlbersEqualArea
plt.figure(figsize=(10,8))
ax = plt.axes(projection=ccrs.AlbersEqualArea())
ax.coastlines();
ax.stock_img();
```



```
# Contoh proyeksi dengan sifat equal area, yakni Sinusoidal
plt.figure(figsize=(10,8))
ax = plt.axes(projection=ccrs.Sinusoidal())
ax.coastlines();
ax.stock_img();
```



```
# Contoh proyeksi dengan sifat equidistant, yakni PlateCarree
plt.figure(figsize=(10,8))
ax = plt.axes(projection=ccrs.PlateCarree())
ax.coastlines();
ax.stock_img();
```



```
# Contoh proyeksi dengan sifat equidistant, yakni AzimuthalEquidistant
plt.figure(figsize=(10,8))
ax = plt.axes(projection=ccrs.AzimuthalEquidistant())
ax.coastlines();
ax.stock_img();
```



Untuk mempelajari lebih lanjut soal proyeksi peta, pembaca disarankan untuk mengunjungi situs:

<https://scitools.org.uk/cartopy/docs/latest/crs/projections>

Namun, jika pembaca masih belum paham kepentingan penggunaan proyeksi yang spesifik, disarankan untuk menggunakan proyeksi *Plate Carrée*.

Menambahkan fitur pada peta

Pada bagian ini kita akan mempelajari cara menambahkan fitur - fitur, seperti garis pantai; perbatasan negara; daratan; danau; sungai; dan laut. Selain itu kita juga akan mempelajari cara menambahkan latarbelakang dan *grid* pada peta, serta memotong peta pada wilayah geografis yang kita kehendaki.

Untuk menambahkan fitur, kita perlu mengimpor modul feature dari pustaka Cartopy.

```
import cartopy.feature as cf

plt.figure(figsize=(10,8))
ax = plt.axes(projection=ccrs.PlateCarree())

# menambahkan fitur garis pantai dengan transparansi 50%
ax.add_feature(cf.COASTLINE, alpha=.5); # bisa juga dgn ax.coastlines()

# menambahkan fitur perbatasan negara dengan transparansi 80% dan style '--'
ax.add_feature(cf.BORDERS, alpha=.8, ls='--');

# menambahkan fitur daratan dengan transparansi 50%
ax.add_feature(cf.LAND, color='#bd7a15', alpha=.5);

# menambahkan fitur danau (yang ditampilkan hanya danau - danau besar saja)
```



```

ax.add_feature(cf.LAKES);

# menambahkan fitur sungai
ax.add_feature(cf.RIVERS);

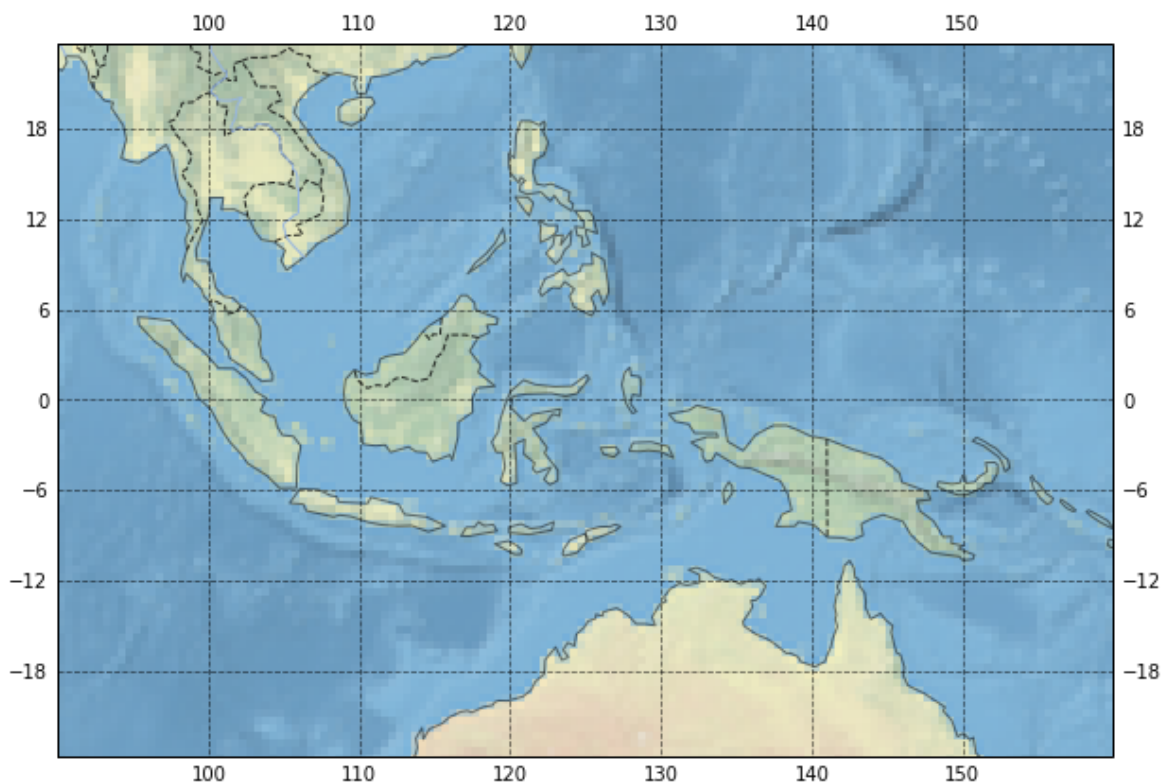
# menambahkan fitur laut
ax.add_feature(cf.OCEAN);

# menambahkan latarbelakang pada peta
ax.stock_img(); # ditambahkan latarbelakang kondisi vegetasi, dll.

# menambahkan grid
ax.gridlines(draw_labels=True, color='black', alpha=.6, linestyle='--');

# memotong bagian Benua Maritim (90 s/d 160; -20 s/d 20)
ax.set_extent([90,160,-20,20])
# [bujurmin, bujurmaks, lintangmin, lintangmaks]

```



Membaca file shp dan menampilkan data geometri negara

Pada bagian ini kita akan mencoba membaca data vektor geografis dalam bentuk *shapefile* secara otomatis dari situs Natural Earth dan mengekstraksi data geometri negara - negara di dunia untuk kemudian kita tampilkan pada peta global. Untuk itu, kita perlu mengimpor dua modul tambahan, yakni *shapereader* (dari modul *io* di pustaka *Cartopy*) dan *patheffects* (dari pustaka *matplotlib*).

```

import cartopy.io.shapereader as shpreader
import matplotlib.patheffects as PE

```

```

plt.figure(figsize=(12,10));

```

```

# menampilkan basemap
ax = plt.axes(projection=ccrs.PlateCarree())
ax.add_feature(cf.COASTLINE, ls='--', alpha=.5);
ax.add_feature(cf.BORDERS, alpha=.5);
ax.add_feature(cf.LAND);
ax.add_feature(cf.OCEAN);
ax.stock_img();

ax.set_extent([-150,60,-25,90]) # menghapus antartika

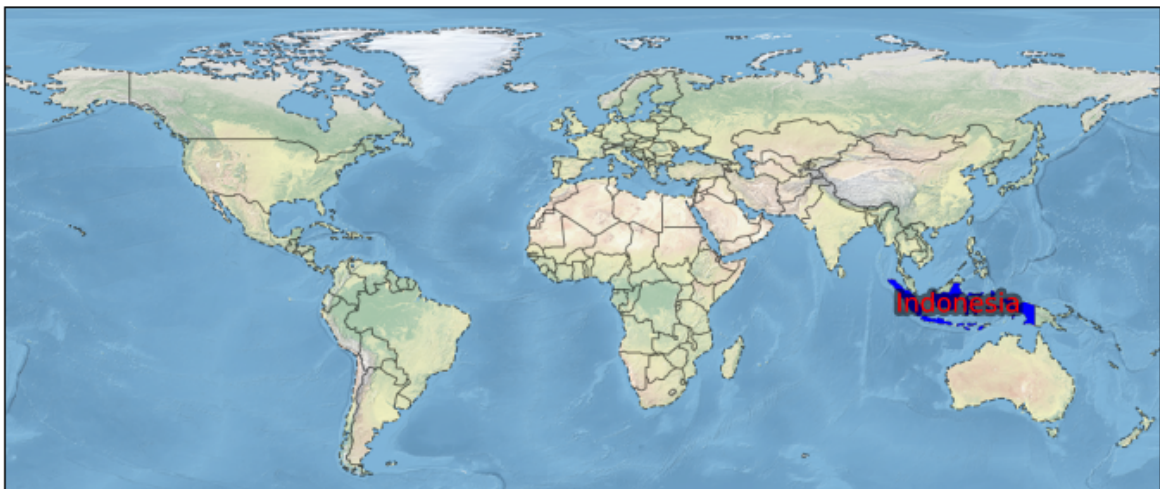
# membaca file shp peta negara dari https://www.naturalearthdata.com/
filesdp = shpreader.natural_earth(resolution='110m',
                                   category='cultural',
                                   name='admin_0_countries')

baca = shpreader.Reader(filesdp) # membaca file shp peta negara - negara

# ekstraksi data geometri negara
negara2 = baca.records()
namaNegara = input('Masukkan Nama Negara: ').title()
for negara in negara2:
    if negara.attributes['NAME'] == namaNegara:
        ax.add_geometries(negara.geometry, ccrs.PlateCarree(),
                           facecolor=(0, 0, 1), label=namaNegara);
        # menambahkan nama negara di bagian tengah negara
        x = negara.geometry.centroid.x
        y = negara.geometry.centroid.y
        ax.text(x,y,negara.attributes['NAME'], color='red', size=15,
                ha='center', va='center', transform=ccrs.PlateCarree(),
                path_effects=
                [PE.withStroke(linewidth=5,foreground='k',alpha=.5)])

```

Masukkan Nama Negara: indonesia



Menampilkan peta gempa hari ini

Sebagai penutup, penyusun akan mengajak sidang pembaca (yang umumnya dari kalangan geosains) untuk melakukan kegiatan pemetaan gempa selama 24 jam terakhir dari data USGS. Diharapkan modul ini akan membantu pembaca untuk mengaplikasikan materi pembelajaran Cartopy pada bidang keilmuannya masing-masing.

Untuk memulai proyek mini ini, kita wajib mengimpor tiga buah pustaka Python, yakni: pandas (untuk membaca data tabular), matplotlib (untuk visualisasi), dan Cartopy (untuk pemetaan).

```
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
```

Selain itu, kita juga perlu mengatur tampilan plot agar tampak lebih estetik.

```
%matplotlib inline
matplotlib.rcParams['figure.figsize'] = (14,10)
```

Kita membaca data tabular secara *remote* dengan menggunakan pandas.

```
df = pd.read_csv('http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/1.0_week.csv')
df.head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	uq
0	2020-07-08T04:37:36.100Z	33.658167	-116.725333	15.36	2.35	ml	83.0	17.00	0.05368	0.20	...	2020-07-08T04:41:3
1	2020-07-08T04:37:10.820Z	36.455833	-117.953500	9.83	1.48	ml	14.0	150.00	0.10340	0.22	...	2020-07-08T04:47:4
2	2020-07-08T04:28:11.830Z	38.136800	-117.888200	9.00	1.30	ml	10.0	90.21	0.02300	0.08	...	2020-07-08T04:34:2
3	2020-07-08T04:14:48.500Z	38.126200	-118.094000	0.30	2.20	ml	24.0	64.54	0.03900	0.23	...	2020-07-08T04:29:3
4	2020-07-08T04:12:46.220Z	36.261333	-89.498497	9.36	2.36	md	26.0	50.00	0.16010	0.21	...	2020-07-08T04:18:1

5 rows × 22 columns

Karena kolom time belum berupa objek datetime, maka kita perlu melakukan konversi sebagai berikut:

```
df['time'] = pd.to_datetime(df['time'])
type(df['time'][1])
```

```
pandas._libs.tslibs.timestamps.Timestamp
```

```
df.head()
```

```

.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}

```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	updated	
0	2020-07-08 04:37:36.100000+00:00	33.658167	-116.725333	15.36	2.35	ml	83.0	17.00	0.05368	0.20	...	2020-07-08T04:41:33.997Z	5 I C
1	2020-07-08 04:37:10.820000+00:00	36.455833	-117.953500	9.83	1.48	ml	14.0	150.00	0.10340	0.22	...	2020-07-08T04:47:40.620Z	1 C F
2	2020-07-08 04:28:11.830000+00:00	38.136800	-117.888200	9.00	1.30	ml	10.0	90.21	0.02300	0.08	...	2020-07-08T04:34:29.366Z	3 C N
3	2020-07-08 04:14:48.500000+00:00	38.126200	-118.094000	0.30	2.20	ml	24.0	64.54	0.03900	0.23	...	2020-07-08T04:29:31.779Z	2 M N
4	2020-07-08 04:12:46.220000+00:00	36.261333	-89.498497	9.36	2.36	md	26.0	50.00	0.16010	0.21	...	2020-07-08T04:18:13.450Z	C V F T

5 rows × 22 columns

Untuk mendapatkan data gempa hari ini, kita perlu melakukan operasi *masking*. Sebagai catatan, Notebook ini dibuat pada tanggal 8 Juli 2020. Oleh karena itu, penyusun akan melakukan *masking* waktu dari tanggal 7 hingga 8 Juli 2020 (hal ini patut disesuaikan oleh pembaca).

```

mask = ((df['time'] >= '2020-07-07') & (df['time'] < '2020-07-08'))
gempaHariIni = df.loc[mask]
gempaHariIni.head()

```

```

.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}

```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	updated	
56	2020-07-07 23:51:01.840000+00:00	19.364666	-155.218506	-0.19	1.89	ml	19.0	113.0	NaN	0.12	...	2020-07-07T23:56:39.490Z	8 k of Vol Ha
57	2020-07-07 23:50:59.200000+00:00	19.355000	-155.219330	-0.99	1.89	ml	13.0	122.0	NaN	0.22	...	2020-07-07T23:56:32.460Z	9 k of Vol Ha
58	2020-07-07 23:49:48.779000+00:00	63.196800	-151.112400	0.00	1.00	ml	NaN	NaN	NaN	0.77	...	2020-07-07T23:52:51.616Z	49 of I Na Par Ala
59	2020-07-07 23:49:28.655000+00:00	12.982600	92.409200	10.00	4.60	mb	NaN	84.0	1.358	1.21	...	2020-07-08T00:19:28.040Z	14 NN Bar Fla Ind
60	2020-07-07 23:31:29.865000+00:00	63.560000	-147.489600	64.70	1.40	ml	NaN	NaN	NaN	0.51	...	2020-07-07T23:36:07.686Z	73 ESE Mc Par Ala

5 rows x 22 columns

Kita dapat mengetahui besaran gempa maksimum dan minimum yang terjadi secara global pada hari ini dengan menggunakan perintah sebagai berikut:

```
print(df[df['mag'] == df['mag'].min()]) # besaran gempa minimum
```

```
      time  latitude  longitude  depth  mag  \
231  2020-07-07 06:30:21.850000+00:00  33.352000 -116.359833  10.90  0.95
260  2020-07-07 03:55:13.260000+00:00  34.126833 -117.478167   5.23  0.95
746  2020-07-05 09:58:50.190000+00:00  37.652500 -118.892333   2.60  0.95
776  2020-07-05 06:58:02.870000+00:00  33.509167 -116.480000  13.75  0.95
880  2020-07-04 21:18:50.110000+00:00  34.465333 -117.966000   8.39  0.95
1415 2020-07-03 04:03:31.650000+00:00  53.860333 -166.751167   8.12  0.95
1449 2020-07-03 02:08:45.240000+00:00  33.580833 -116.801667   7.10  0.95
1586 2020-07-02 17:50:28.090000+00:00  37.461833 -118.727500   4.54  0.95
1781 2020-07-02 05:51:54.490000+00:00  33.334500 -116.187333   8.11  0.95

magType  nst  gap  dmin  rms  ...  updated  \
231      ml  34.0  68.0  0.043300  0.23  ...  2020-07-07T06:34:03.667Z
260      ml  15.0 123.0  0.097480  0.13  ...  2020-07-07T03:58:59.157Z
746      md  19.0  87.0  0.008846  0.08  ...  2020-07-06T17:02:03.770Z
776      ml  36.0  63.0  0.064330  0.17  ...  2020-07-06T14:26:40.280Z
880      ml  19.0  55.0  0.018550  0.08  ...  2020-07-07T20:34:43.089Z
1415     ml   6.0 102.0  0.044930  0.09  ...  2020-07-06T18:12:24.430Z
1449     ml  38.0  28.0  0.033880  0.21  ...  2020-07-03T15:07:33.230Z
1586     md  15.0 156.0  0.151300  0.03  ...  2020-07-02T18:51:05.078Z
1781     ml  29.0 120.0  0.106000  0.24  ...  2020-07-02T05:55:50.184Z

      place  type  horizontalError  \
231  11km N of Borrego Springs, CA  earthquake      0.33
260      4km NNW of Fontana, CA  earthquake      0.45
746      8km ENE of Mammoth Lakes, CA  earthquake      0.38
776      19km ESE of Anza, CA  earthquake      0.24
880      6km SSE of Littlerock, CA  earthquake      0.14
1415  14 km WSW of Dutch Harbor, Alaska  earthquake      0.35
1449      12km WNW of Anza, CA  earthquake      0.27
1586      12km SSW of Toms Place, CA  earthquake      0.46
1781      17km SSW of Oasis, CA  earthquake      0.39

depthError  magError  magNst  status  locationSource  magSource
231         0.56      0.158    24.0  automatic          ci          ci
260         1.24      0.324    18.0  automatic          ci          ci
746         0.29      0.254    15.0  reviewed           nc          nc
776         0.45      0.180    28.0  reviewed           ci          ci
880         0.25      0.153    13.0  reviewed           ci          ci
1415        0.62      0.245     6.0  reviewed          av          av
1449        0.74      0.151    27.0  reviewed           ci          ci
1586        2.63      0.245    13.0  reviewed           nc          nc
1781        1.82      0.124    21.0  automatic          ci          ci

[9 rows x 22 columns]
```

```
print(df[df['mag'] == df['mag'].max()])
```

```
      time  latitude  longitude  depth  mag  \
328  2020-07-06 22:54:46.856000+00:00  -5.6368   110.6783  528.66  6.6

magType  nst  gap  dmin  rms  ...  updated  \
328     mw  NaN  20.0  1.422  0.92  ...  2020-07-07T23:00:40.068Z

      place  type  horizontalError  depthError  \
328  93 km N of Batang, Indonesia  earthquake      8.3      6.4

magError  magNst  status  locationSource  magSource
328      0.068    21.0  reviewed           us          us

[1 rows x 22 columns]
```

Sesudah itu, kita akan mengekstraksi data bujur, lintang, dan besaran gempa (dalam skala Richter) dalam bentuk objek *list*:

```
bujur = list(df['longitude'])
lintang = list(df['latitude'])
besaran = list(df['mag'])
```

Kemudian kita akan mengklasifikasikan titik - titik gempa dengan menggunakan warna - warna tertentu (hijau untuk gempa di bawah 3 SR, kuning untuk gempa dengan rentang 3 - 5 SR, dan merah untuk gempa di atas 5 SR) dengan menggunakan fungsi sebagai berikut:

```
def warna(besaran):
    if besaran < 3.0:
        return 'g'
    elif 3.0 <= besaran < 5.0:
        return 'y'
    else:
        return 'r'
```

Kemudian kita tinggal melakukan pemetaan dengan menggunakan Cartopy:

```
ax = plt.axes(projection = ccrs.PlateCarree())
ax.coastlines(resolution='50m')
ax.stock_img()

for i in range(len(besaran)):
    warnaEpi = warna(besaran[i])
    plt.scatter(bujur[i], lintang[i], s=besaran[i]*10, c=warnaEpi)

plt.title('Peta gempa bumi global pada 7 - 8 Juli 2020');
```

