

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

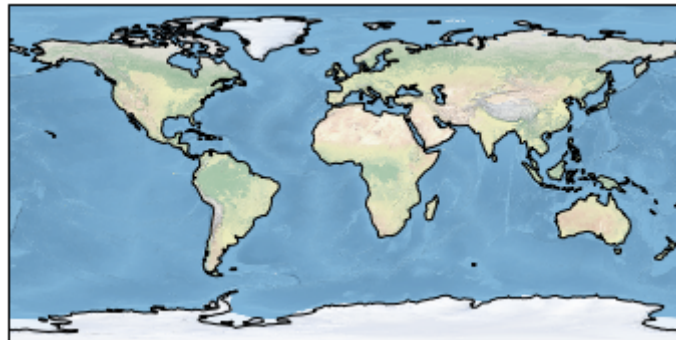
`plt.show()` setiap kali hendak menampilkan grafik.

```
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
```

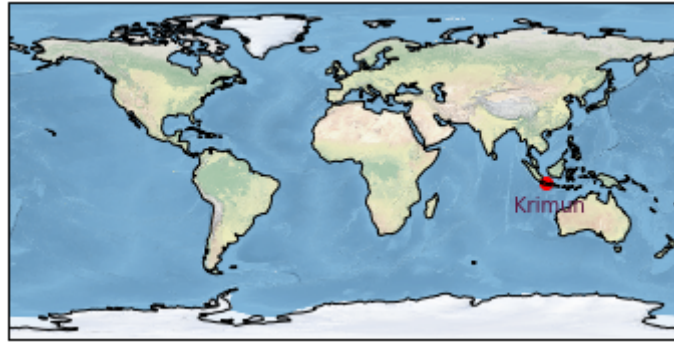


Kemudian kita akan mencoba menampilkan lokasi penyusun bermukim saat ini di Desa Krimun, Losarang, Indramayu, Jawa Barat (108.157691, -6.388166) dengan menggunakan *scatterplot*:

```
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');
```

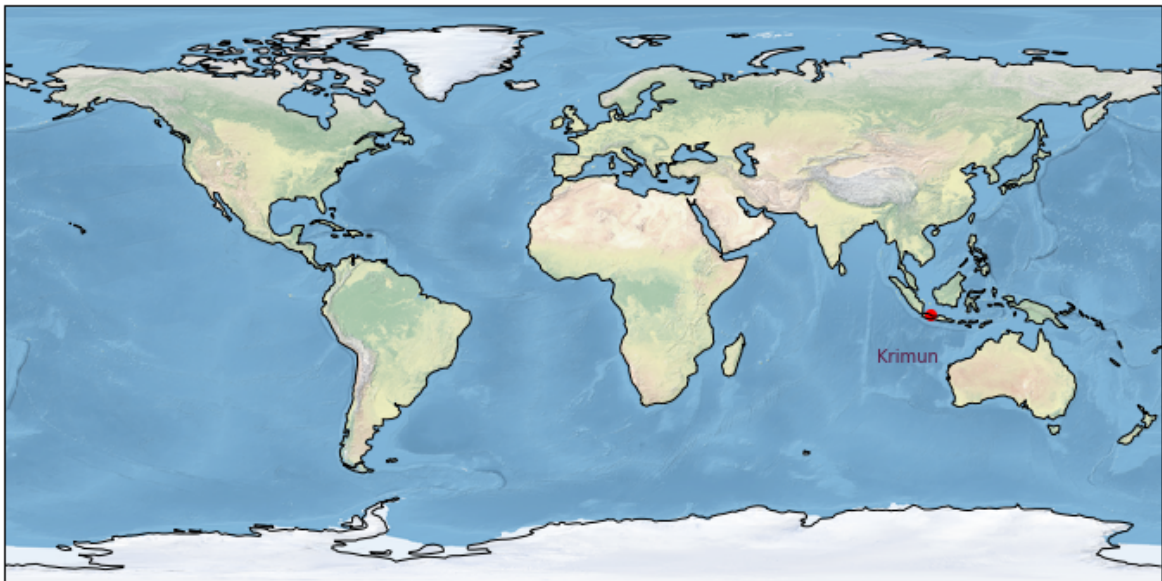


Untuk mengatur ukuran peta, kita dapat menggunakan metode `figure()` di modul `pyplot` di bagian awal *script* dengan menambahkan argumen `figsize` dalam bentuk *tuple* (lebar, tinggi) dalam satuan inci dalam kasus ini ukuran peta adalah 12×10 inci:

```
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

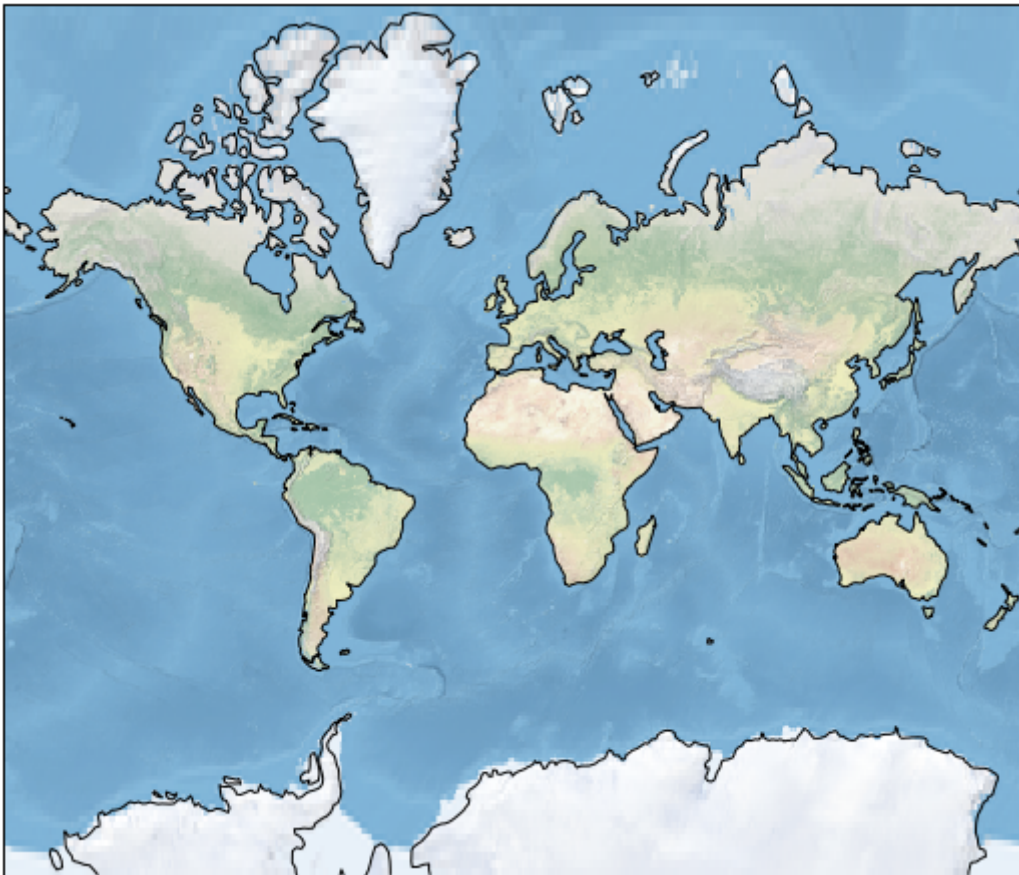
Ketika menampilkan permukaan bumi yang bersifat 3D ke lembar peta 2D, maka pasti terjadi distorsi. Oleh karena itu, untuk mengurangi distorsi tersebut terdapat berbagai macam proyeksi yang dapat digunakan. Suatu peta yang ideal harus memenuhi ketiga syarat sebagai berikut:

- *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).

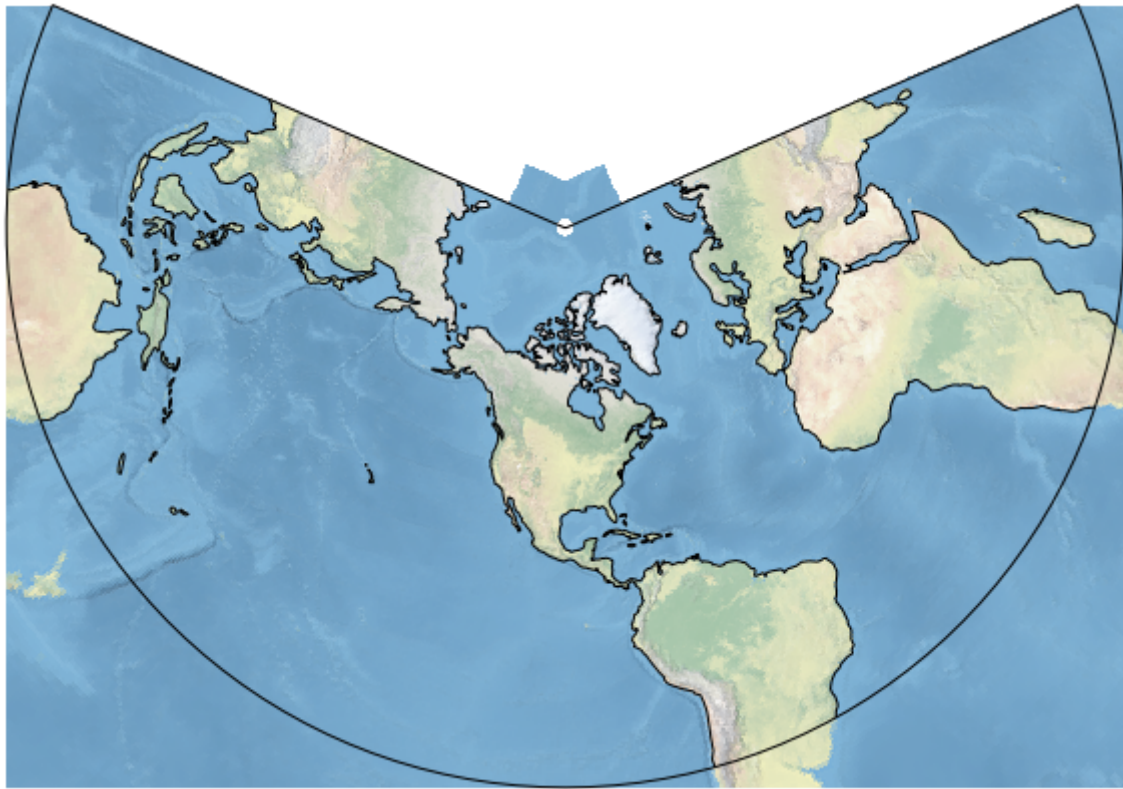
- *Equal area* : Proyeksi yang bersifat *equal area* merupakan proyeksi yang merujuk pada luasan yang sama antara di peta dengan rupa bumi aslinya (namun terdapat distorsi pada bentuk, sudut, dan/atau skalanya).
- *Equidistant*: Proyeksi yang bersifat *equidistant* adalah proyeksi yang mempertahankan 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.

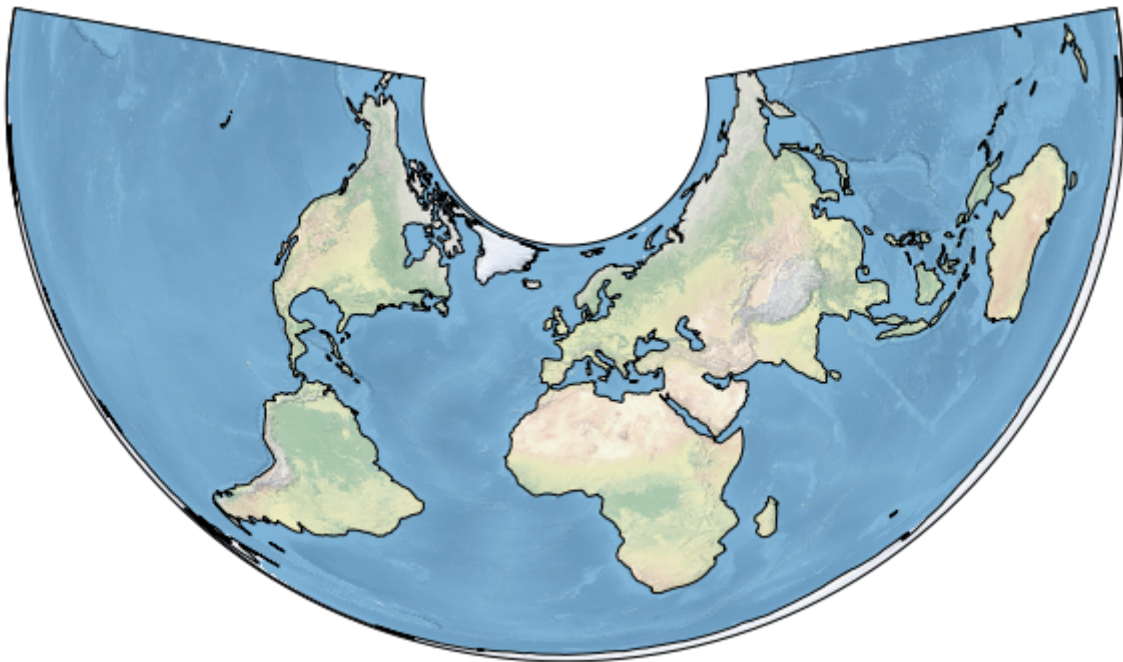
```
# Contoh proyeksi dengan sifat conformal, yakni Mercator
plt.figure(figsize=(10,8))
ax = plt.axes(projection=ccrs.Mercator())
ax.coastlines();
ax.stock_img();
```



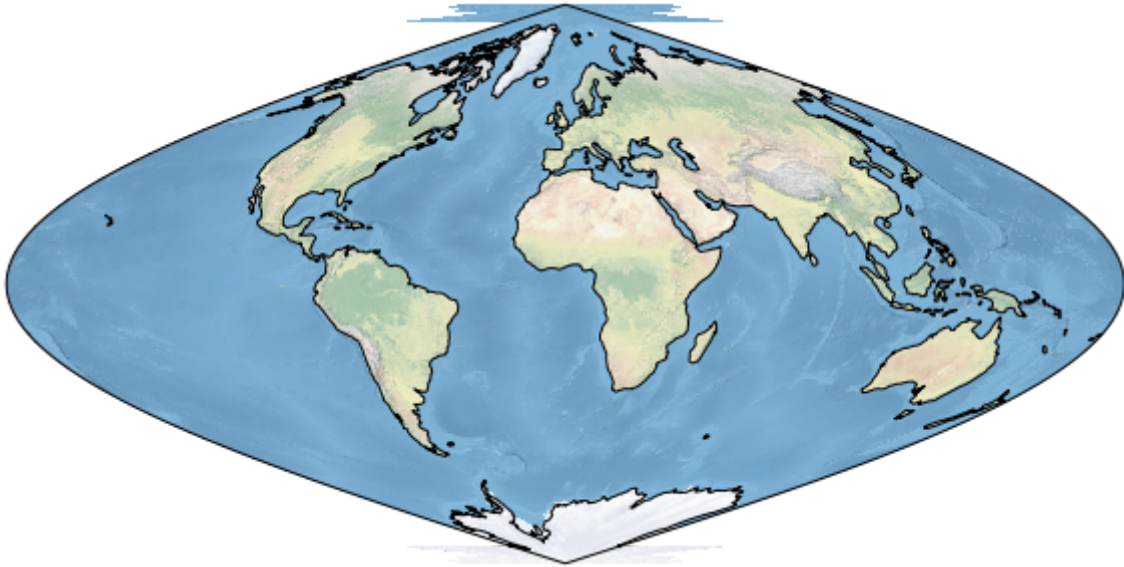
```
# Contoh proyeksi dengan sifat conformal, yakni LambertConformal
plt.figure(figsize=(10,8))
ax = plt.axes(projection=ccrs.LambertConformal())
ax.coastlines();
ax.stock_img();
```

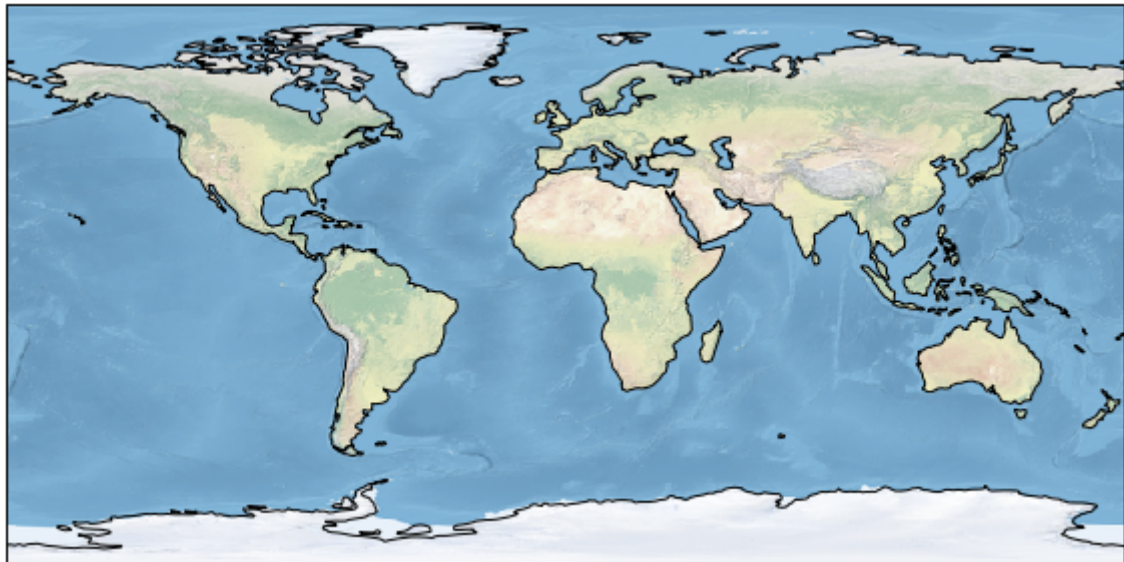
```
# Contoh proyeksi dengan sifat equal area, yakni AlbersEqualArea
plt.figure(figsize=(10,8))
ax = plt.axes(projection=ccrs.AlbersEqualArea())
ax.coastlines();
ax.stock_img();
```



```
# 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);

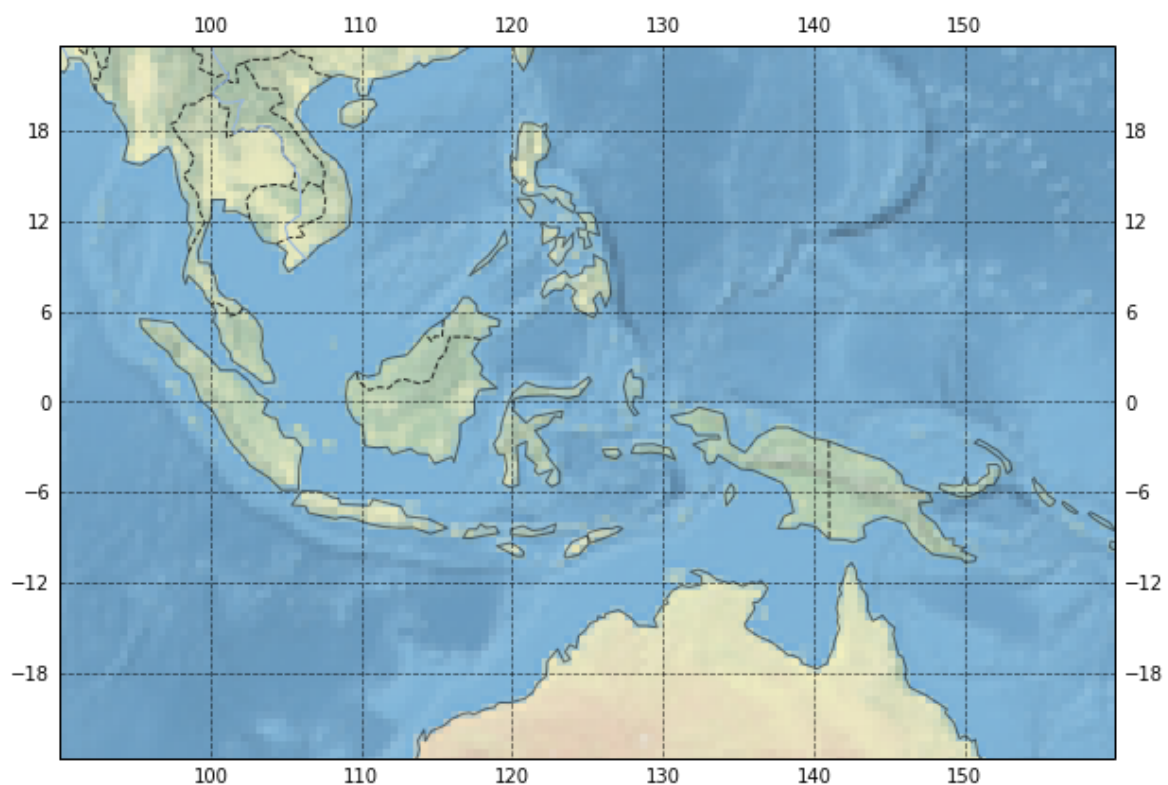
# menambahkan fitur laut
ax.add_feature(cf.OCEAN);

# 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

