

**MODUL PRAKTIKUM INTERAKSI ATMOSFER DAN LAUT
MODUL DASAR PENGOLAHAN DI IDL**

MODUL PRAKTIKUM



Koordinator Mata Kuliah :

Dr. Sc. Anindya Wirasatriya, ST, MSi, MSc

**DEPARTEMEN OSEANOGRAFI
FAKULTAS PERIKANAN DAN ILMU KELAUTAN
UNIVERSITAS DIPONEGORO
SEMARANG
2021**

MODUL PRAKTIKUM

Modul I. Download Data

1.1. OISST

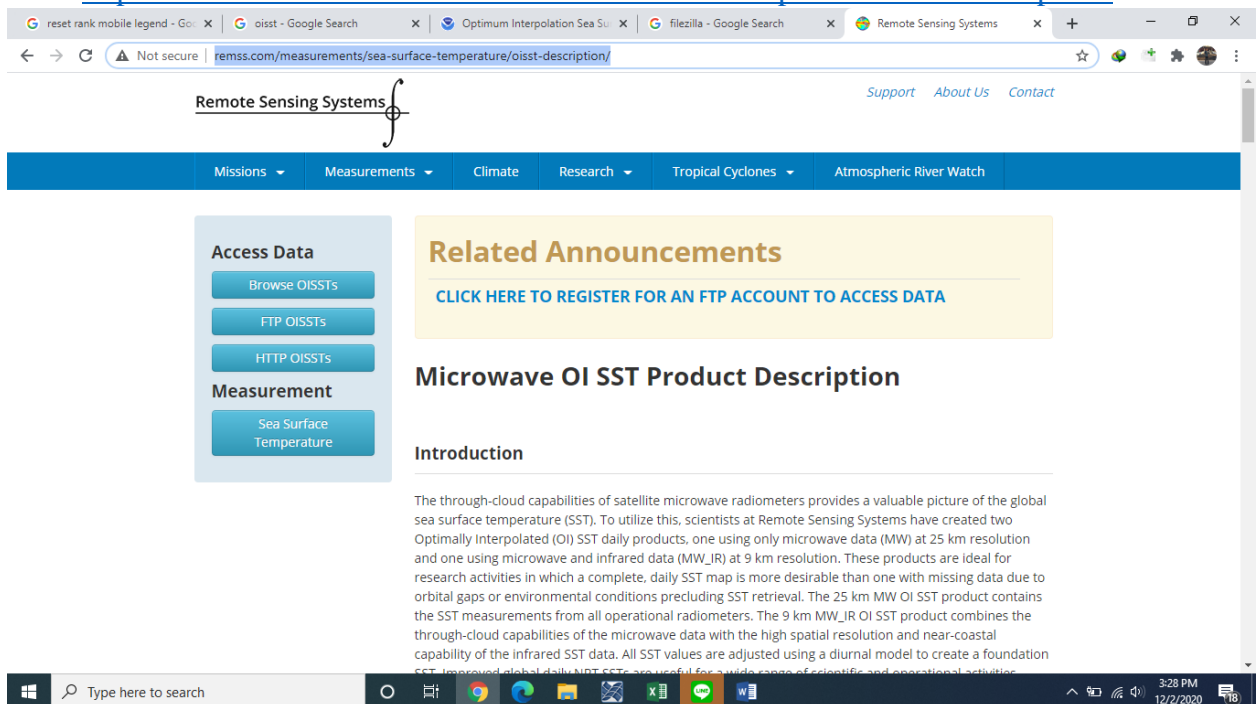
Optimum Interpolation Sea Surface Temperature (OISST) adalah analisis yang dibangun dengan menggabungkan pengamatan dari berbagai platform (satelit, kapal, pelampung, dan pelampung Argo) pada jaringan global reguler. Peta SST yang lengkap secara spasial dibuat dengan melakukan interpolasi untuk mengisi celah. Metodologi tersebut mencakup penyesuaian bias pengamatan satelit dan kapal (mengacu pada pelampung) untuk mengkompensasi perbedaan platform dan bias sensor. Ini terbukti kritis selama Mt. Letusan Pinatubo tahun 1991, ketika tersebar luasnya aerosol vulkanik mengakibatkan suhu satelit infra merah jauh lebih dingin dari suhu laut sebenarnya (Reynolds, 1993).

1.2. Aplikasi FileZilla

Aplikasi FileZilla adalah aplikasi gratis (*free software*) yang digunakan untuk membuka dan membuat halaman ftp. FTP (*File Transfer Protocol*) adalah internet service yang dirancang untuk membuat sambungan ke server internet tertentu atau komputer, sehingga user dapat mengirimkan file ke komputer (download) atau mengirimkan file ke server (upload). FTP saat ini banyak digunakan untuk melakukan pertukaran data, karena lebih mudah daripada menggunakan perangkat kabel atau fisik.

1.3. Download Data

1. Buka halaman REMSS OISST terlebih dahulu. Untuk dapat mengakses dapat menggunakan link : <http://www.remss.com/measurements/sea-surface-temperature/oisst-description/>



Remote Sensing Systems

Missions Measurements Climate Research Tropical Cyclones Atmospheric River Watch

Access Data

- Browse OISSTs
- FTP OISSTs
- HTTP OISSTs

Measurement

- Sea Surface Temperature

Related Announcements

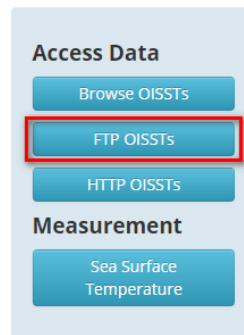
[CLICK HERE TO REGISTER FOR AN FTP ACCOUNT TO ACCESS DATA](#)

Microwave OI SST Product Description

Introduction

The through-cloud capabilities of satellite microwave radiometers provides a valuable picture of the global sea surface temperature (SST). To utilize this, scientists at Remote Sensing Systems have created two Optimally Interpolated (OI) SST daily products, one using only microwave data (MW) at 25 km resolution and one using microwave and infrared data (MW_IR) at 9 km resolution. These products are ideal for research activities in which a complete, daily SST map is more desirable than one with missing data due to orbital gaps or environmental conditions precluding SST retrieval. The 25 km MW OI SST product contains the SST measurements from all operational radiometers. The 9 km MW_IR OI SST product combines the through-cloud capabilities of the microwave data with the high spatial resolution and near-coastal capability of the infrared SST data. All SST values are adjusted using a diurnal model to create a foundation SST. Improved global daily NDT SSTs are useful for a wide range of scientific and operational activities.

2. Daftar terlebih dahulu untuk mendapat akun REMSS OISST
3. Pilih FTP OISST



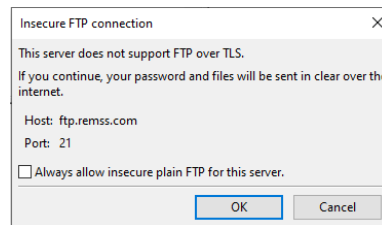
4. Copy link yang muncul pada Tab Browser

`ftp://ftp.remss.com/sst/daily/`

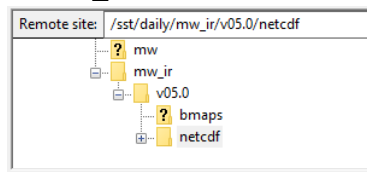
5. Buka FileZilla dan Paste di bagian Host lalu isi Username dan Password, untuk Port dikosongkan saja dan klik Quickconnect



6. Bila ada Popup yang muncul klik OK

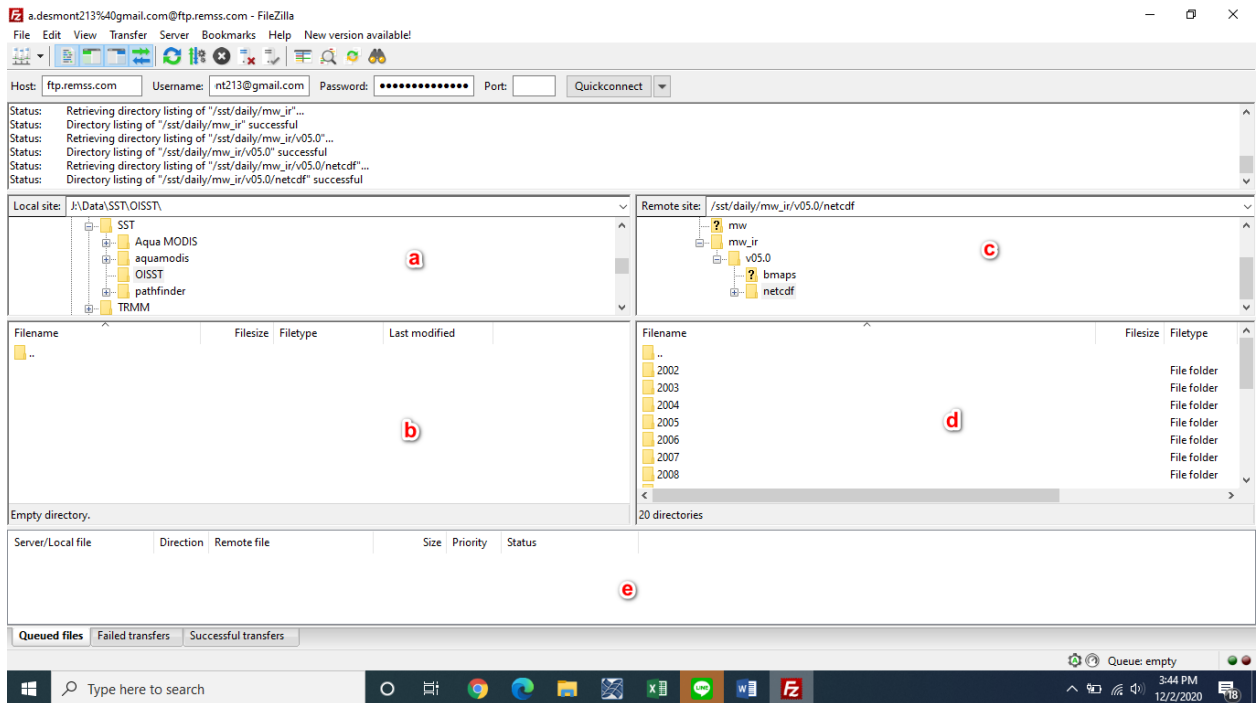


7. Klik dan masuk pada kotak kanan mw_ir > v5.0 > netcdf

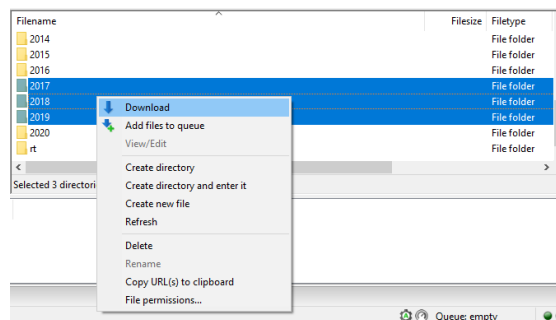


8. Perhatikan gambar berikut :





- a. Kotak ini digunakan untuk memilih folder penyimpanan di komputer kita
 - b. Kotak ini digunakan untuk mengecek file-file yang sudah terdownload/tersedia di komputer kita
 - c. Kotak ini digunakan untuk memilih folder pada server yang akan kita download
 - d. Kotak ini digunakan untuk memilih data yang akan di download
 - e. Kotak ini berfungsi untuk monitoring file yang sedang di download
9. Pada kotak d tadi, pilih data yang akan di download (usahakan minimal 3 tahun berurutan), lalu klik kanan dan download

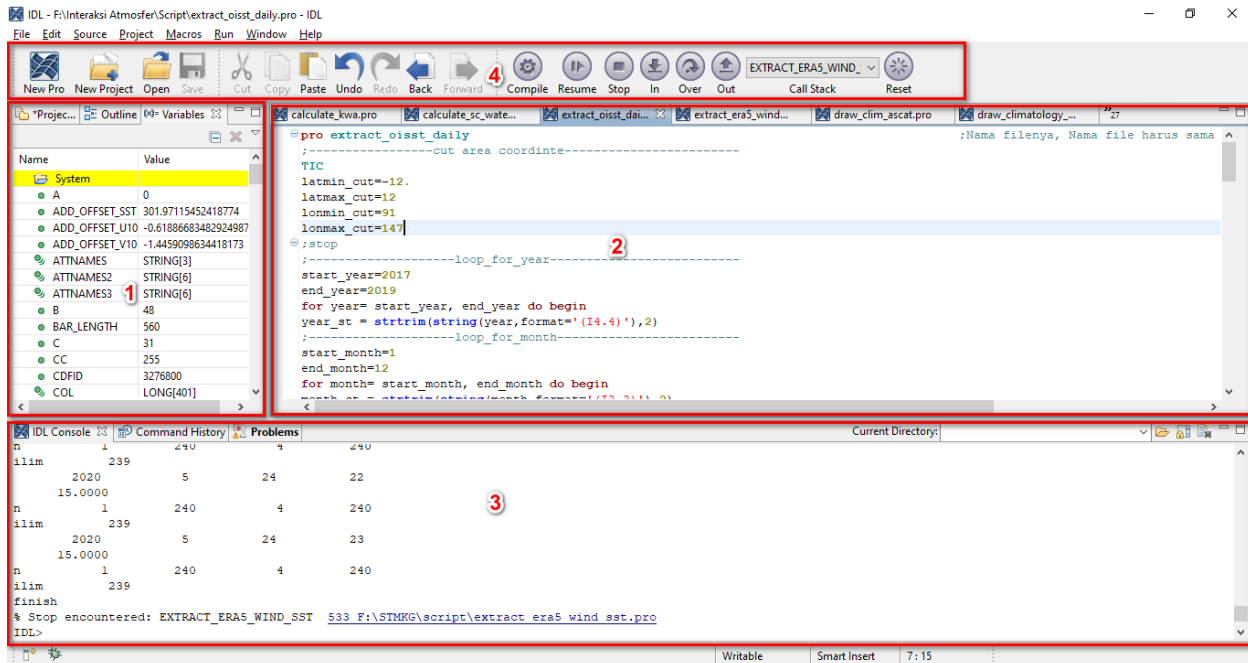


Modul II. GUI IDL

2.1. IDL

IDL, singkatan dari Interactive Data Language, adalah bahasa pemrograman yang digunakan untuk analisis data. Ini populer di bidang sains tertentu, seperti astronomi, fisika atmosfer, dan pencitraan medis. IDL memiliki sintaks yang sama dengan PV-Wave dan berasal dari basis kode yang sama, meskipun bahasanya kemudian berbeda secara detail. Ada juga implementasi gratis atau tanpa biaya, seperti GNU Data Language (GDL) dan Fawlt Language

2.2. GUI IDL



1. Variable Tabs = Tempat Variable disimpan dari hasil running program
2. Text Editor = Tempat untuk melakukan pengkodean
3. Console = Tempat untuk menampilkan hasil program
4. Menu Bar
 - a. New Pro = Membuat script baru
 - b. Open = membuka file pro (*.pro) atau file sav (*.sav)
 - c. Save = menyimpan script ke dalam file pro
 - d. Cut = memotong
 - e. Copy = Menyalin
 - f. Paste = Tempel
 - g. Undo = Mengembalikan hasil edit script
 - h. Redo = Membatalkan Undo
 - i. Back = Kembali ke file pro sebelumnya
 - j. Forward = Membatalkan Back
 - k. Compile = Mengecek seluruh file agar siap dirunning
 - l. Run = Menjalankan script yang telah dibuat



- m. Resume = Melanjutkan running script yang telah diberi “stop” pada script atau di hentikan dengan Pause
- n. Stop = menghentikan running program
- o. Pause = menghentikan sementara running program dan masih bisa dilanjutkan dengan resume
- p. Reset = mereset kembali script supaya tidak ada data yang tersangkut



Modul III. Ekstrak Data

3.1. Ekstrak Data >> (ekstrak_oisst_daily.pro)

Aturan dasar dalam penulisan script di IDL harus selalu dimulai dengan “pro” dan diakhiri dengan “end”. Setelah menulis pro disebelahnya diikuti dengan nama file tersebut. Contoh sebagai berikut:

```
extract_oisst_daily 8/29/2020 5:55 PM >>> File Pro yang digunakan
pro extract_oisst_daily >>> Memulai program dan nama file program
end >>> Mengakhiri program
```

```
=====
latmin_cut=-12.
latmax_cut=12
lonmin_cut=91
lonmax_cut=147
;stop
```

Pada script tersebut digunakan untuk memotong data sesuai dengan wilayah yang diinginkan.

stop = umumnya digunakan untuk menghentikan program dengan tujuan untuk mengecek program diatas stop sudah sesuai dengan yang diinginkan atau belum

```
=====
;stop
;-----loop_for_year-----
start_year=2017
end_year=2019
for year= start_year, end_year do begin
year_st = strtrim(string(year,format='(I4.4)'),2)
;-----loop_for_month-----
start_month=1
end_month=12
for month= start_month, end_month do begin
month_st = strtrim(string(month,format='(I2.2)'),2)
;=====
start_day=1
end_day=31
for day= start_day, end_day do begin
day_st = strtrim(string(day,format='(I2.2)'),2)

endfor
endfor
endfor
```

Looping (for) pada IDL :

Format looping pada IDL seperti pada contoh diatas

for **variable** = **variable mulai** , **variable akhir** do begin
endfor



for do begin = digunakan untuk memulai rangkaian looping

endfor = digunakan untuk mengakhiri rangkaian looping

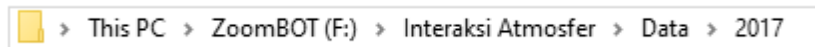
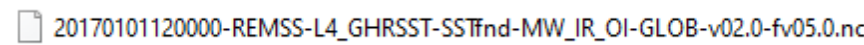
string = digunakan untuk mengubah atau mengkonversi suatu karakter (integer, float, dex, binner)
menjadi bentuk string (bentuk huruf dan karakter)

strtrim = digunakan untuk menghapus space yang dihasilkan oleh string

```
'      56' >>> Hasil string
```

```
'56' >>> Hasil strtrim dan string
```

```
=====
path_input='F:\Interaksi Atmosfer\data\' + year_st + '\' + year_st + month_st + day_st + '*.nc'
datanc= findfile(path_input, count=num_files)
=====
```

 >>> Alamat folder >>> Nama file

Untuk memasukkan file ke path_input, alamat file harus berupa string

PENTING:

- Perlu diperhatikan alamat file harus sama dengan alamat pada path_input
- Untuk memisahkan antar folder digunakan tanda “\”
- Tanda “*” digunakan untuk membaca semua nama file yang tidak dicantumkan pada path_input
- findfile = digunakan untuk mengubah tanda “*” menjadi nama file yang tidak dicantumkan tadi
- count = digunakan untuk menghitung jumlah data dari findfile dan dimasukan ke dalam “num_files”

```
if num_files gt 0 then begin
for inc=0,num_files-1 do begin

endfor
endif
```

Percabangan pada IDL dapat menggunakan IF dan diakhiri dengan ENDIF

Format IF pada IDL :

```
if Variabel kondisi Variabel_kondisi then begin
endif
```

Variabel = ditempati oleh data



Kondisi = dapat diisi dengan:

- gt = Greater Then / lebih dari
- ge = Greater Then Equivalent / lebih dari sama dengan
- eq = Equivalent / sama dengan
- ne = Not Equivalent / tidak sama dengan
- lt = Less Then / kurang dari
- le = Less Then Equivalent / kurang dari sama dengan

Variabel_kondisi = diisi dengan syarat untuk memenuhi kondisi tersebut

Contoh : mencari data di Variabel **SST** yang **lebih dari 29°C** akan ditambah 1

If **SST** **gt** **29** then begin

SST = SST + 1

Endif

```
=====
;=====
cdfid = NCDF_OPEN(datanc[inc])
```

NCDF_OPEN = digunakan untuk membuka file NetCDF dan dimasukkan kedalam variable “cdfid”

datanc[inc] = mengambil data dari variable “datanc” urutan ke-“inc”

```
varnames = ncdf_vardir(cdfid)
;stop
NCDF_VARGET, cdfid, 0, lat
NCDF_VARGET, cdfid, 2, time
NCDF_VARGET, cdfid, 3, analysed_sst
NCDF_VARGET, cdfid, 1, lon

IDL> print, varnames
lat lon time analysed_sst analysis_error sea_ice_fraction mask
0> 1 2 3 4 5 6
```

Ncdf_vardir = digunakan untuk mencari data apa saja yang tersedia (variable yang tersedia) pada file netcdf

Ncdf_varget = digunakan untuk mengambil data yang tersedia tadi untuk dimasukkan kedalam variable yang diinginkan



Format pengambilan data :

NCDF_VARGET, Variabel dari NCDF_OPEN, urutan / kolom data yang mau di ambil, variable

Info : Urutan pada IDL dimulai dari 0, 1, 2, dst.

```
attnames = ncdf_attdir(cdfid, 'analysed_sst')
```

Ncdf_attdir = digunakan untuk mengetahui attribute apa saja yang ada di dalam suatu variabel, dalam kasus ini mencari attribute pada “analysed_sst”

Penting : variable yang akan di cari attributnya harus sesuai dengan yang ada pada varnames

```
NCDF_ATTGET, cdfid, 'analysed_sst', '_FillValue', _FillValue
NCDF_ATTGET, cdfid, 'analysed_sst', 'add_offset', add_offset
NCDF_ATTGET, cdfid, 'analysed_sst', 'units', units
NCDF_ATTGET, cdfid, 'analysed_sst', 'scale_factor', scale_factor
```

```
IDL> print, attnames
_FillValue units long_name standard_name add_offset scale_factor valid_min valid_max source
IDL> |
```

NCDF_ATTGET = digunakan untuk mengambil attribute tertentu pada suatu variable (file NetCDF) dan dimasukkan kedalam variable (IDL) tertentu

```
lon = FLOAT[4096]
-179.956 -179.868 -179.780 -179.692 -179.604 -179.517
-179.429 -179.341 -179.253 -179.165 -179.077 -178.989
-178.901 -178.813 -178.726 -178.638 -178.550 -178.462
-178.374 -178.286 -178.198 -178.110 -178.022 -177.935
-177.847 -177.759 -177.671 -177.583 -177.495 -177.407
-177.319 -177.231 -177.143 -177.056 -176.968 -176.880
-176.792 -176.704 -176.616 -176.528 -176.440 -176.352
-176.265 -176.177 -176.089 -176.001 -175.913 -175.825
-175.737 -175.649 -175.561 -175.474 -175.386 -175.298
-175.210 -175.122 -175.034 -174.946 -174.858 -174.770
-174.683 -174.595 -174.507 -174.419 -174.331 -174.243
Press 'F2' for focus
```

```
sst=shift(analysed_sst, -2048)
```

Shift(analysed_sst, -2048) = digunakan untuk menggeser data dari analysed_sst sebanyak 2048 data ke kiri

Digeser supaya nilai Longitude 0° berada pada awal data (untuk memudahkan dalam pengkodean kedepannya)



```
;=====
latmin=min(lat)
lonmin=0.0439453
grid_interval=lon[1]-lon[0]
sst_map=sst*scale_factor+add_offset
;=====
```

Min(lat) = digunakan untuk mencari nilai minimal pada variable lat

Lon[1] = mengambil data urutan ke-2 pada longitude

Lon[0] = mengambil data urutan pertama pada longitude

Tanda “*” digunakan untuk mengalikan

Tanda “+” digunakan untuk menambahkan

Perlu dikalikan dengan scale factor dan add offset dikarenakan pada file sst (analysed_sst) nilainya masih berupa nilai integer, harus dikalikan dengan scale_factor terlebih dahulu dan ditambahkan juga dengan add_offset supaya nilai dari sst tersebut menjadi nilai suhu yang benar.

```
;=====
index = where(sst eq _FillValue, count_index)
;stop
sst_map[index]=9999.
```

Where : digunakan untuk menyeleksi data. Dalam kasus ini, seleksi digunakan untuk mencari nilai error (_FillValue) dan jumlah data nilai error tadi dihitung pada count_index

Sst_map[index]=9999 : Artinya memasukkan nilai 9999 ke sst_map pada lokasi yang telah diseleksi dari where tadi

```
-
lon_beg = lonmin_cut
lon_end = lonmax_cut
lat_beg = latmin_cut
lat_end = latmax_cut
a = fix ((lon_beg-lonmin)/grid_interval)
b = fix ((lon_end-lonmin)/grid_interval)
c = fix ((lat_beg-latmin)/grid_interval)
d = fix ((lat_end-latmin)/grid_interval)

data_map = sst_map[a:b,c:d]
```

Pada script ini digunakan untuk memotong data agar sesuai dengan longitude dan latitude yang kita masukkan diawal.



Contoh kasus

10 Longitude										4 Latitude
11	12	13	14	15	16	17	18	19	20	
21	22	23	24	25	26	27	28	29	30	
31	32	33	34	35	36	37	38	39	40	
41	42	43	44	45	46	47	48	49	50	

Jika Indonesia berada pada longitude 6 – 9 dan latitude 3 – 4 maka di program menyeleksi dengan
Data_map=sst_map[5:8,2:3]

Jadi hasilnya :

4 Longitude				2 Latitude
36	37	38	39	
46	47	48	49	

```
=====
file_mkdir,'F:\Interaksi Atmosfer\Olah\Ekstrak2\' + year_st + '\'+ month_st + '
save,data_map,filename='F:\Interaksi Atmosfer\Olah\Ekstrak2\' + year_st + '\'+ month_st + '\sst'+year_st+month_st+day_st+'.sav'
:stop
NCDF_CLOSE, Cdfid
```

File_mkdir = digunakan untuk membuat folder, perlu karena dalam save data nanti, script “save” tidak bisa membuat folder sendiri dan folder harus sudah tersedia terlebih dahulu

Save = digunakan untuk menyimpan data dalam format *.sav

Format penulisan save

Save, **Variable yang akan disimpan**, filename= '**Alamat penyimpanan folder/Namafilename.sav**'

NCDF_CLOSE = digunakan untuk menutup file NetCDF yang telah dibuka dengan NCDF_OPEN

```
=====
endfor
endif
endfor
endfor
endfor
:stop
TOC
print, 'finish'
stop
end
```

Penutup For dan IF yang dibuat tadi

Print digunakan untuk menampilkan data pada Console, dalam hal ini menampilkan kata finish

End untuk menutup program, tanda program telah selesai



Modul IV. Compile Data Monthly

4.1. Compile Data Monthly >> (compile_monthly_oisst.pro)

Pada Compile Data Monthly ini berfungsi untuk merata-ratakan data harian menjadi data bulanan. Hal ini biasa dilakukan dalam penelitian dikarenakan data bulanan yang paling mudah untuk dilihat dan dianalisis baik dalam bentuk sebaran spasial maupun dalam bentuk grafik jika data tersebut merupakan data timeseries yang cukup panjang.

```
grid_interval=lon[1]-lon[0]  
grid_interval = 0.087890625 Factor+add  
Press 'F2' for focus
```

>> Skript Ekstrak

```
data_map = sst_map[a:b,c:d]  
data_map = DOUBLE[639, 275]  
300.58000 300.54000 300.51000 300.47000 300.42000  
300.35000 300.33000 300.33000 300.34000 300.27000  
300.32000 300.32000 300.31000 300.31000 300.35000  
300.37000 300.38000 300.39000 300.38000 300.39000  
300.43000 300.46000 300.43000 300.44000 300.48000  
300.51000 300.52000 300.56000 300.48000 300.47000  
300.51000 300.69000 300.63000 300.42000 300.42000  
300.41000 300.52000 300.58000 300.66000 300.59000  
300.56000 300.54000 300.54000 300.41000 300.38000  
300.44000 300.51000 300.83000 300.84000 300.69000  
300.63000 300.62000 300.66000 300.76000 300.79000
```

> Data yang mau disave di Ekstrak

Dipake untuk image_width dan image_height (Dimensi Data)

```
pro compile monthly oisst  
TIC  
;=====1  
grid interval= 0.087890625 2  
latmin=-12  
latmax=12 3  
lonmin=91  
lonmax=147 4  
image_width = round((lonmax-lonmin)/grid_interval+2)  
image_height = round((latmax-latmin)/grid_interval+2)  
;stop  
;=====vear=====  
image_width = 639 image_height = 275  
Press 'F2' for focus Press 'F2' for focus
```

>Compile Monthly

1. Mulai script dan nama file (penjelasan seperti pada Ekstrak)
2. grid_interval = didapat dari script Ekstrak > Resolusi Data
3. Area (latmin,lonmin,latmax,lonmax) harus sama seperti crop pada ekstrak



4. image_width dan image_height harus sama seperti Dimensi Data pada Ekstrak. Jika berbeda disesuaikan saja (+2 diubah jadi +1 atau dihapus atau -1 atau lainnya)

```
;=====year=====
start_year=2017
end_year=2019

for year= start_year, end_year do begin
year_st = strtrim(string(year,format='(I4.4)'),2)
;=====month=====
start_month=1
end_month=12
for month= start_month, end_month do begin
month_st = strtrim(string(month,format='(I2.2)'),2)
;
```

Sama seperti Ekstrak, disini sesuai dengan rentang waktu yang digunakan. Hanya sampai Bulan (month) karena pada skrip Compile Monthly berfungsi untuk merata-ratakan data selama 1 bulan

```
;=====
SST monthly map = fltarr(image width,image height) 1
num_monthly_map = fltarr(image_width,image_height) 2
num=0.
>STOP
```

> dari Script

$$\frac{1}{n} (x_1 + x_2 + \dots + x_n)$$

> Rumus Kompilasi

1. Digunakan untuk menjumlahkan data (di kasus ini menjumlahkan data suhu)
2. Digunakan untuk menunjukkan banyaknya data (jika 1 bulan maka jumlahnya antara 28-31)

Fltarr = digunakan untuk membuat matrix baru sebanyak image_width,image_height

```
restore,datanc[inc]
```

Digunakan untuk membuka file dari hasil ekstrak (khusus untuk penyimpanan menggunakan syntax “save” seperti berikut)

```
save,data_map,filename='F:\Interaksi Atmosfer\Olah\Ekstrak2\' + year_st + \'\' + month_st + \'\' + sst + year_st + month_st + day_st + \'.sav'
```

```
indx_sst = where (data_map ne 999., count_index)
;stop
=====

SST_monthly_map[indx_sst] = SST_monthly_map[indx_sst] + data_map[indx_sst]
num_monthly_map[indx_sst] = num_monthly_map[indx_sst] + 1
num=num+1

;stop
endfor
endif
```

Setelah data dibuka menggunakan “restore” data dari save (ekstrak) dapat digunakan dengan memanggil variable sesuai saat save data ekstrak > “data_map”. Kemudian data di pilih menggunakan “where” agar nilai suhu yang 999 (error) tidak ikut terhitung.

Seperti diatas, di dalam script ini merupakan perhitungan untuk mengumpulkan data 1 bulan terlebih dahulu

Contoh sederhana

Data SST 3 hari akan dirata2kan :

Hari 1 = 29

Hari 2 = 28.5

Hari 3 = 28

Jika berdasarkan script itu maka,

$SST_MONTHLY_MAP = 29 + 28.5 + 28$

$NUM_MONTHLY_MAP = 3$

```
indx_sst_monthly=where(num_monthly_map gt 0,count_indx_sst_monthly)
data_map=fltarr(image_width,image_height)+999.
;stop
data_map[indx_sst_monthly]=SST_monthly_map[indx_sst_monthly]/float(num_monthly_map[indx_sst_monthly])
```

Pada script ini berfungsi untuk perhitungan rata-rata setelah mengetahui jumlah data dan banyaknya data

$data_map=fltarr.....+999$ > berarti membuat matrix baru dengan seluruh isi matrixnya 999

```
file_mkdir, 'F:\Interaksi Atmosfer\Olah\Compile Monthly\'+year_st+' '
path_output='F:\Interaksi Atmosfer\Olah\Compile Monthly\'+year_st+'\sst'+year_st+month_st+'.sav
save, DATA_MAP, filename=path_output
```

Menyimpan data, Sama seperti Ekstrak. Disini hanya sampai bulan karena data harian telah di satukan atau dikompilasi menjadi data bulanan sehingga selama 1 tahun hanya ada 12 data saja.



Modul V. Compile Data Climatology

5.1. Compile Data Climatology >> (compile_climatology_oisst.pro)

Pada Compile Data Climatology ini, kita merata-ratakan data selama beberapa tahun menjadi 1 data (dibulan yang sama) misalkan data dari tahun 2000 – 2020, pada kompilasi klimatologi ini yang dilakukan adalah merata-ratakan data pada bulan januari dari tahun 2000 – 2020 kemudian dilanjutkan pada bulan febuari dari tahun 2000 – 2020 hingga bulan desember dari tahun 2000 – 2020. Pada akhirnya Compile Data Climatology ini akan menghasilkan 12 data saja (Januari – Desember). Umumnya Climatology ini digunakan untuk melihat kondisi dalam keadaan normal dari beberapa tahun olahan karena bisa jadi kejadian seperti ENSO, IOD dan lainnya mengakibatkan kenaikan atau penurunan data pada tahun tahun tertentu.

```
pro compile_climatology_oisst
TIC
;=====
grid_interval= 0.087890625

latmin=-12
latmax=12
lonmin=91
lonmax=147

image_width = round((lonmax-lonmin)/grid_interval+2)
image_height = round((latmax-latmin)/grid_interval+2)
```

Seperti biasa, pro untuk mulai, compile_climatology_oisst merupakan judul scriptnya dan nama file.

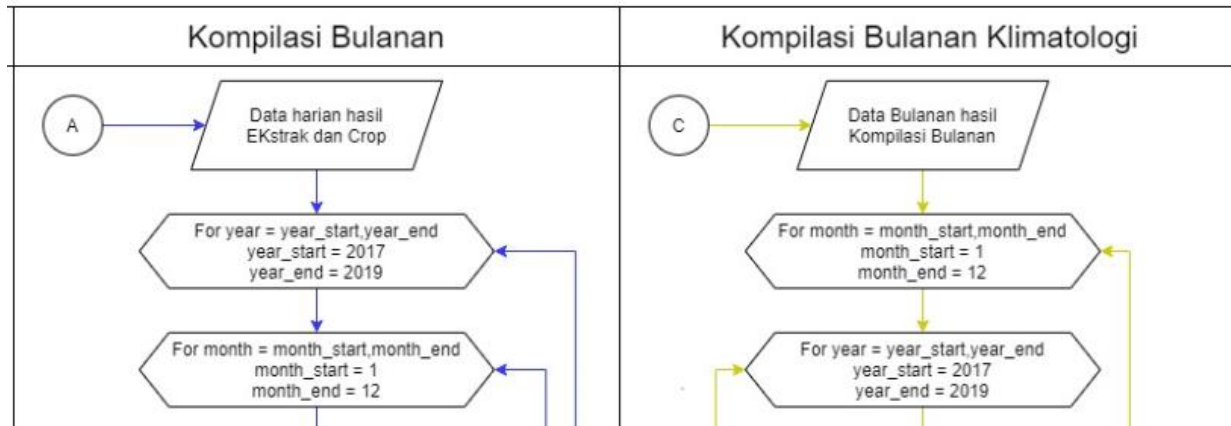
Grid_interval masih sama seperti di Ekstrak dan Compile Monthly

Lokasi Penelitian juga harus sama seperti sebelumnya (Ekstrak dan Compile Monthly)

Image_width dan image_height juga disamakan dengan Dimensi data (Ekstrak dan Compile Monthly)

NB: Disini masih sama dan harus sama dengan Compile Monthly, jika ada yang berbeda maka disesuaikan lagi dan dicek kembali antara nilai maksimal dan minimalnya





Disini perbedaan antara Kompilasi Bulanan dan Kompilasi Bulanan Klimatologi. Pada Compile Monthly, loopingnya dimulai dari looping tahun kemudian masuk looping bulan tetapi pada Compile Climatology dimulai dari looping bulan kemudian masuk looping tahun. Jika Compile Monthly hasilnya 12 per tahun (jadi jika 3 tahun ada 36 data), Compile Climatology menghasilkan data 12 saja selama beberapa tahun (lebih dari 3 tahun).

```

start_month=1
end_month=12
for month= start_month, end_month do begin
month_st = strtrim(string(month,format='(I2.2)'),2)
=====
SST_monthly_map = fltarr(image_width,image_height)
num_monthly_map = fltarr(image_width,image_height)
num=0.
=====year=====
start_year=2017
end_year=2019

for year= start_year, end_year do begin
year_st = strtrim(string(year,format='(I4.4)'),2)
=====

```

1. Rentang waktu penelitian yang digunakan. Disini dimulai dari looping bulan dan dilanjutkan dengan looping tahun
2. Pembuatan matrix tetap pada looping bulan seperti pada compile monthly dikarenakan setiap bulan kita perlu reset data matrix agar bulan selanjutnya tidak tercampur dengan bulan sebelumnya.

```

/
path_input='F:\Interaksi Atmosfer\Olah\Compile Monthly\' + year_st + '\sst' + year_st + month_st + '.sav'
datanc= findfile(path_input,count=num_files)
;stop
if num_files gt 0 then begin
for inc=0,num_files-1 do begin

restore,datanc[inc]

```

Input data dari hasil Compile Monthly.

```
SST_monthly_map[indx_sst] = SST_monthly_map[indx_sst] + data_map[indx_sst] ; Data yang kurang dari 100 masuk perhitungan
num_monthly_map[indx_sst] = num_monthly_map[indx_sst] + 1
num=num+1

;stop
endfor
endif
endfor
;=====AVERAGING (default)=====
indx_sst_monthly=where(num_monthly_map > 0, count_indx_sst_monthly)
data_map=filtarr(image_width,image_height)+999.
;stop
data_map[indx_sst_monthly]=SST_monthly_map[indx_sst_monthly]/float(num_monthly_map[indx_sst_monthly]) ;float = numerik desimal
;stop
```

Sama seperti Compile Monthly, disini digunakan untuk menghitung rata – rata data yang dipakai.

```
file_mkdir, 'F:\Interaksi Atmosfer\Olah\Compile Climatology\'
path_output='F:\Interaksi Atmosfer\Olah\Compile Climatology\sst'+month_st+'.sav
save,data_map,filename=path_output
endfor
;stop
print, 'finish'
TOC
stop
end
```

Script ini untuk menyimpan data. Karena datanya hanya 12 maka hanya memerlukan month saja.



Modul VI. Draw Climatology

6.1. Draw Climatology

Pada script ini kita akan menggambarkan distribusi spasial data klimatologi bulanan yang sudah di jalankan pada script compile climatology.

```
function true2lon,rgb
;rgb is array of lonarr(3)
return, (long(rgb[0]) or ishft(long(rgb[1]),8) or ishft(long(rgb[2]),16))
end
;=====
```

Function true2lon, satu paket script ini digunakan untuk menggambarkan warna yang sudah di atur pada script sebelumnya kedalam canvas. Karena script ini berupa function, maka script ini dapat berada diluar pro dan hanya digunakan ketika dipanggil menggunakan true2lon.

```
;=====
pro draw_climatology_oisst
;=====
!p.background='ffffff'x
!p.color=0
;=====
!P.Region=[0.02,0.09,0.98,0.98]
!P.Position=[0.07,0.07,0.95,0.98]
;=====
```

Seperti biasa, script harus dibuka dengan pro diikuti dengan nama scriptnya.

- !p.background = digunakan untuk mewarnai background dari canvas. 'ffffff'x = warna putih
- !p.color = digunakan untuk memberi warna tulisan, 0 = warna hitam
- !p.Region = digunakan untuk memberi layout
- !p.Position = digunakan untuk meletakkan posisi peta yang akan digambarkan (ini akan menyesuaikan sesuai dengan luasan wilayah yang digunakan)

```
;=====
loadct,33,/silent
tvlct,r,g,b,/get
rgb1=(long(r) or ishft(long(g),8) or ishft(long(b),16))
rgb2=[[rgb1 and 'ff'x],[[ishft(rgb1 and 'ff00'x,-8)],[[ishft(rgb1 and 'ff0000'x,-16)]]]
;=====
xsize=1000
ysize=720
;=====
bar_length=fix(xsize*0.7)
rgb3=byte(congrid(rgb2,bar_length,15,3))
;stop
```

- loadct = digunakan untuk memuat warna, 33 = kode warna dari color table ke-33. Untuk kode warnanya dapat dilihat di Google : Color Table IDL.
- tvlct = digunakan untuk memecah kode warna dari loadct tadi
- xsize dan ysize = ukuran pixel (canvas) yang akan digunakan
- bar_lenght = panjang bar yang akan digambar
- rgb3 = membuat warna pada bar yang akan digambar
- rgb1 = memasukkan warna kedalam bentuk matrix sesuai dengan r,g,b
- rgb2 = menggabungkan data dari rgb1 menjadi data 3 dimensi



```

;=====data coverage=====
;=====indonesia=====
latmin = -12
latmax = 12
lonmin= 91
lonmax= 147
;=====image coverage=====
;=====indonesia=====
latmin_plot = -12
latmax_plot = -5
lonmin_plot= 100
lonmax_plot= 120
;=====
;=====spatial_resolution=====
grid_interval=0.087890625
;=====data_dimension=====
image_width=round((lonmax-lonmin)/grid_interval+2)
image_height=round((latmax-latmin)/grid_interval+2)
;stop

```

- Data coverage (latmin, lonmin, latmax, lonmax) = Area yang digunakan saat cropping (seperti sebelum-sebelumnya), disini harus sama dengan saat cropping.
- Image coverage = Area yang akan digambarkan pada script draw (tidak harus sama dengan data coverage tetapi harus masuk dalam area data coverage)
- grid_interval = resolusi data yang digunakan (dilihat pada saat ekstrak)
- data_dimension = dimensi data yang digunakan (ini harus sama dengan saat ekstrak, compile monthly, compile climatology), bila ada yang berbeda maka disesuaikan lagi (+nya)

```

;=====Looping Month=====
start_month=1
end_month=12
for month=start_month,end_month do begin
month_st=strtrim(string(month,format='(I2.2)'),2)
;=====Data Input=====
path_input='F:\Praktikum Pemrograman Oseanografi\Pengolahan\Compile Climatology\sst'+month_st+'.sav'
;=====Open Data=====
restore,path_input
data=data_map:-273.15
idx= where(data lt 500.)
;stop

```

- Looping data bulanan karena datanya hanya 12 bulan saja
- Path_input = alamat dan nama file yang akan di input. Di path_input ada '+month_st+' disini akan berubah berubah sesuai dengan bulan yang dijalankan.
- restore = membuka file yang disimpan dengan menggunakan save
- untuk memanggil datanya, variable yang digunakan harus sama dengan saat menyimpan. Disini menggunakan data_map

```

;=====
window,0,xsize=xsize,ysize=ysize;,/pixmap
;stop
lon2=lonmin+findgen(round(image_width))*grid_interval
lat2=latmin+findgen(round(image_height))*grid_interval
;stop
;=====
map_set,0,180,limit=[latmin_plot,lonmin_plot,latmax_plot,lonmax_plot],/iso
;=====

```

- window,0, = digunakan untuk membuat sebuah canvas kosong yang nantinya akan digunakan untuk menggambarkan petanya.
- Lon2 = lonmin..... = disini longitude yang tidak disimpan saat ekstrak tadi dibuat. Hal ini digunakan untuk meletakkan posisi tiap data sst yang akan di gambarkan kedalam peta. Sama halnya dengan lat2 untuk membuat latitude seperti di lon2.
- map_set = disini computer akan men-setting area yang telah ditentukan pada Image coverage tadi



```
=====
;stop
max_value=31
min_value=25
index_max=where((data gt max_value)and(data lt 500),count_max)
if (count_max ne 0)then data[index_max]=max_value

index_min=where(data lt min_value,count_min)
if (count_min ne 0)then data[index_min]=min_value
;stop
=====
```

Pada dasarnya computer hanya akan menggambarkan sesuai dengan apa yang user perintahkan. Pada script ini user memberikan perintah untuk menggambarkan SST hanya dari suhu 25 – 31 °C berarti suhu yang kurang dari 25 atau lebih dari 31 tidak akan digambarkan oleh computer sehingga perlu di manipulasi untuk suhu yang dibawah 25 itu diganti dengan 25 dan suhu yang lebih tinggi dari 31 (bukan nilai error / 999) diganti menjadi 31. Hal ini tidak akan mengubah intepretasi pada peta.

```
=====Color Interval=====
level_interval=0.05
numlevel=fix((max_value-min_value)/level_interval+1)
col=lonarr(numlevel)
;stop
for e=0,numlevel-1 do begin
  cc=byte(e*level_interval/(max_value-min_value)*255)
  col[e]=true2lon(rgb2(cc,0,*))
endfor
levels=findgen(numlevel)*level_interval+min_value
;stop
```

Color Interval digunakan untuk mengatur gradasi warna yang akan digunakan. Jika semakin kecil level_intervalnya maka semakin bagus gradasi warnanya hanya saja untuk merunning seluruh datanya membutuhkan waktu yang lama. Di script ini true2lon tadi dipanggil untuk memberikan warna pada data.

```
contour,data,lon2,lat2,/overplot, $
  max_value=max_value, $
  min_value=min_value, $
  levels=levels, $
  /cell_fill, $
  font=1, $
  c_colors=col[indgen(numlevel)]
;stop
```

Contour disini digunakan untuk meletakkan data SST dengan warna-warnanya tadi sesuai dengan longitude dan latitude yang telah ada.

```
=====
map_continents,/coasts,mthinethick=2.5,/ hires
;stop
map_grid,/box_axes,londel=3., latdel=3.,charsize=2.5,charthick=2.5
;stop
=====
```

- map_continents = digunakan untuk memberikan basemap pada peta. Disini menggunakan /coasts yang berarti akan menggambarkan garis pantainya.
- Map_grid = digunakan untuk membuat grid-grid peta. Londel dan latdel digunakan untuk memberikan interval pada longitude dan latitude untuk gridnya.



```
tv,rgb3,tr=3,0.15,0.14,/normal
;stop
;=====interval value in colorbar=====
interval_measure=1
num_interval=round((max_value-min_value)/interval_measure)
for w=0,num_interval do begin
    index_value=w*(max_value-min_value)/float(num_interval)+min_value
    index_plot=string(index_value,format='(F5.1)')
    x_position=0.15+0.7/num_interval*w
    y_position=0.09
    xyouts,x_position,y_position,index_plot,charsize=3.,charthick=3.,font=1,/normal,alignment=0.5
endfor
```

- Tv,rgb3..... = digunakan untuk mendisplay bar yang tadi sudah disiapkan diawal, 0.15 merupakan posisi Xnya dan 0.14 merupakan posisi Y
- Interval value in colorbar digunakan untuk memberikan keterangan interval data yang digunakan. Interval_measure merupakan interval dari keterangan yang akan dibuat.
- X_position = merupakan posisi x (0.15)
- Y_position = merupakan posisi y

```
title2='SST '+month_st
XYOUTs,0.5,0.9,title2,/normal,alignment=0.5,charsize=3.5,charthick=1.5,font=1,color=255
XYOUTs,0.5,0.19,'Longitude (!U0!NE)',/normal,alignment=0.5,charsize=3.,charthick=2.5,font=1,color='000000'x1
XYOUTs,0.03,0.5,'Latitude (!U0!NN)',/normal,alignment=0.5,charsize=3.,charthick=2.5,font=1,orientation=90,color='000000'x1
unit_st='(!U0!NC)'
XYOUTs,0.93,0.09,unit_st,/normal,alignment=0.5,charsize=3.,charthick=2.0,font=1,color=255
```

Disini digunakan untuk menambahkan keterangan tambahan seperti judul, longitude, latitude, dan satuan. Dengan menggunakan xyout supaya dapat ditampilkan pada canvas dan dilanjutkan untuk posisi X dan posisi Y. pengaturan tambahan seperti :

- Charsize = mengatur ukuran karakter
- Chartick = mengatur ketebalan karakter
- Orientation = mengatur kemiringan karakter

```
;=====Save Data=====
file_mkdir, 'F:\Praktikum Pemrograman Oseanografi\Hasil\Climatology\Selatan Jawa
path_output ='F:\Praktikum Pemrograman Oseanografi\Hasil\Climatology\Selatan Jawa\sst_'+month_st+'.png'
T=TVRD(channel=0,true=1,order=0)
write_png,path_output,T
;=====
```

- File_mkdir = membuat folder
- Path_output = tempat meletakkan file yang akan disimpan dan memberikan nama filenya
- T = TVRD..... = digunakan untuk menggambarkan pada file png
- Write_png..... = digunakan untuk menuliskan kedalam file png



Modul VII. Convert OISST Climatology to Timeseries

7.1. Convert OISST Climatology to Timeseries

Script ini digunakan untuk mengkonversi data climatology (12 file) menjadi data timeseries (1 file) yang berisi 12 bulan.

```
=====
pro convert_climatology_oisst_timeseries
=====
TIC
=====data coverage=====
=====indonesia=====
latmin=-12
lonmin=91.
latmax=12.
lonmax=147.
=====
=====spatial_resolution=====
grid_interval=0.087890625
=====data_dimension=====
image_width=round((lonmax-lonmin)/grid_interval+2)
image_height=round((latmax-latmin)/grid_interval+2)
count_msst=0
;stop
```

Seperti sebelum – sebelumnya script dimulai dengan pro, lalu data coverage diatur sesuai dengan wilayah saat cropping, grid_interval yang disamakan juga dan data dimension juga disamakan.

```
=====Looping Month=====
start_month=1
end_month=12
for month=start_month,end_month do begin
month_st=strtrim(string(month,format='(I2.2)'),2)
;=====Data Input=====
path_input3='F:\Praktikum Pemrograman Oseanografi\Pengolahan\Compile Climatology\sst'+month_st+'.sav';
;=====Open Data=====
restore,path_input3
sst_map=data_map;-273.15
;stop
```

Sama seperti draw, karena menggunakan data climatology maka looping hanya bulan saja. Alamat file dan nama file diletakkan pada path_input3 dan dibuka dengan restore. Data dipanggil menggunakan data_map karena saat compile climatology datanya disimpan dengan data_map




```
=====case1 natuna
latmin_cut1=5
lonmin_cut1=107
latmax_cut1=8
lonmax_cut1=110
=====case2 selatan jawa
latmin_cut2=-11
lonmin_cut2=108
latmax_cut2=-10
lonmax_cut2=112
===== crop data case1
a1 = fix ((lonmin_cut1-lonmin)/grid_interval)
b1 = fix ((lonmax_cut1-lonmin)/grid_interval)
c1 = fix ((latmin_cut1-latmin)/grid_interval)
d1 = fix ((latmax_cut1-latmin)/grid_interval)
;stop
sst_map21= sst_map[a1:b1,c1:d1]
;stop
===== crop data case2
a2 = fix ((lonmin_cut2-lonmin)/grid_interval)
b2 = fix ((lonmax_cut2-lonmin)/grid_interval)
c2 = fix ((latmin_cut2-latmin)/grid_interval)
d2 = fix ((latmax_cut2-latmin)/grid_interval)
;stop
sst_map22= sst_map[a2:b2,c2:d2]
```

Seperti saat melakukan cropping pada ekstrak, di script ini juga melakukan cropping data untuk menentukan lokasi yang akan dibuat grafiknya. Script ini mengambil 2 wilayah di case1 dan case2. Sebenarnya disini tidak harus 2 wilayah dapat berupa 1 wilayah, 2 wilayah, 3 wilayah dan seterusnya.

```
===== Average case1
;stop
indx_sst1=where(sst_map21 lt 999.0)
Sst_mean1=mean(sst_map21[indx_sst1])
;stop
===== Average case2
indx_sst2=where(sst_map22 lt 999.0)
Sst_mean2=mean(sst_map22[indx_sst2])
=====
```

Data yang sudah di cropping pada script diatas akan diubah menjadi 1 data dengan menggunakan rata-rata (mean). Sebelumnya dilakukan filtering terlebih dahulu untuk nilai error agar nilai error tidak masuk perhitungan rata-ratanya.

```
if(count_msst eq 0) then begin
dataset1=[sst_mean1,sst_mean2,month]
endif else begin
dataset1=[[dataset1],[sst_mean1,sst_mean2,month]]
endelse
;stop
count_msst=count_msst+1
;stop
endfor
```

Pada script ini, data perbulan tadi diubah menjadi data timeseries. Dimana urutan kolomnya harus disesuaikan dengan data yang diconvert ke dalam timeseries. Pada script convert ini menggunakan 2 wilayah (case1 dan case2) dan juga waktu (bulan) sehingga menggunakan 3 kolom untuk datanya (SST case1, SST case2 dan bulan).

```
=====Save Data=====
file_mkdir, 'F:\Praktikum Pemrograman Oseanografi\Pengolahan\Timeseries'
path_output='F:\Praktikum Pemrograman Oseanografi\Pengolahan\Timeseries\timeseries.txt'
format1='(3F15.5)'
openw,3,path_output1
printf,3,dataset1, format=format1
close,3
=====
```

- File_mkdir = membuat folder
- Path_output = alamat penyimpanan dan nama file



- c. Format1 = merupakan format yang akan digunakan untuk penulisan pada file txt. (3F15.5) artinya, 3 = 3 kolom yang digunakan tadi, F merupakan Float (Bilangan decimal), 15 merupakan 15 karakter yang boleh dituliskan pada text nanti, .5 = 5 angka di belakang koma
 - d. Openw = untuk membuka file txt / membuat file txt dan menuliskan file kedalamnya (open and write)
 - e. Printf = untuk menuliskan data ke dalam txt tadi. Yang dituliskan berupa variable dataset dengan format yaitu format1
 - f. Close = untuk menutup file yang tadi dibuka di openw (supaya filenya tidak menyangkut di computer)
-



Modul VIII. Plot Timeseries OISST

8.1. Plot Timeseries OISST

Setelah melakukan convert tadi, maka data yang telah jadi timeseries dapat digambarkan menjadi grafik menggunakan script `plot_time_series_oisst.pro`.

```
=====
> pro plot_time_series_oisst
>
=====background_color=====
!p.background='ffffff'x
!p.color=0
=====window region=====
!P.Region=[0.01,0.09,0.85,0.94]
>
=====window pixel=====
xsize=800
ysize=800
```

Seperti biasa, dimulai dengan `pro` dan nama scriptnya. Dan mirip saat `draw`, fungsi `background` untuk memberikan warna background pada canvas, `color` untuk memberikan warna tulisan, `region` untuk memberikan layout. Dan `window pixel` untuk memberikan ukuran pixel canvasnya.

```
=====input_data=====
path_input_all = 'F:\Praktikum Pemrograman Oseanografi\Pengolahan\Timeseries\timeseries.txt'
PRINT, 'path_input_all: ', path_input_all
>
=====reading_ascii_data (default)=====
;stop
openr, 1, path_input_all
Each_record_sst_all = ftnarr (3) ;=====number of columns=====
count_sst_all = 0
; stop
WHILE NOT EOF (1) DO BEGIN
  READF, 1, Each_record_sst_all
;stop
  if (count_sst_all eq 0) then begin
    Record_sst_all = Each_record_sst_all
    count_sst_all = count_sst_all+1
  ; stop
  endif else begin
    Record_sst_all = [[Record_sst_all], [Each_record_sst_all]]
  ; stop
  endwhile
ENDWHILE
;stop
close, 1
```

- Path_input_all = untuk meletakkan alamat dan nama file txt hasil convert tadi.
- Openr = digunakan untuk membuka dan membaca file txt (open and read), diikuti dengan alamat dan nama file di path_input_all
- Sebelum masuk lebih lanjut, perlu mengatur jumlah kolom yang digunakan saat convert, karena menggunakan 3 kolom maka pada `ftnarr(x)` ditulis dengan 3 ($x = 3$) menjadi `ftnarr(3)`
- While not EOF (1) do begin = digunakan untuk melooping data hingga datanya habis
- Readf = membaca data per baris. Data perbaris akan dibuat menjadi seperti dataset di script convert
- Close = file txt yang di open di tutup.

```
=====
sst_all = size(Record_sst_all, /dimension)
num_data_sst_all = sst_all[1]
kolom=sst_all[0]
baris=sst_all[1]
print, num_data_sst_all
>stop
=====dataset_array=====
y1 = transpose(record_sst_all[0,*]);Natuna
y2 = reform (record_sst_all[1,*], num_data_sst_all);Selatan Jawa

x = reform (record_sst_all[2,*], num_data_sst_all)
>stop
=====x&y_axis_value=====
xrange=[1,12]
yrange=[25,31]
>stop
```



- Size = digunakan untuk mengetahui dimensi dari data, dalam kasus ini data record_sst_all.
- Num_data_all, kolom, dan baris = digunakan untuk mengambil dimensi tertentu, [0] urutan pertama, [1] urutan ke -2 dari data dimensinya
- Transpose = digunakan untuk mengubah data kolom menjadi baris atau sebaliknya. Hal ini dilakukan untuk membuat data 12 baris menjadi data 1 baris (data array)
- Reform, sama seperti transpose
- Xrange = untuk membuat range data X
- Yrange = membuat range data Y

```
=====
date_label=label_date[date_format=['%M','%Y']]
color_array=['000000'x, 'ff0000'x,'00ff00'x,'0099ff'x,'0000ff'x,'990000'x,'009966'x, '7f7f7f'x,'db70db'x, 'ffa300'x]
=====
window,0,xsize=xsize,ysize=ysize
;stop
```

- Label_date = digunakan untuk membuat label berupa data waktu, dalam kasus ini membuat label dalam bulan dan tahun. (umumnya script ini digunakan untuk data timeseries yang panjang seperti data timeseries selama 3 tahun (bukan klimatologi))
- Color_array = berisi kumpulan kode warna untuk plot grafik nanti
- Window = seperti sebelumnya untuk membuat canvas dengan ukuran xsize dan ysize.

```
MONTH=['Ja', 'Fe', 'Ma', 'Ap', 'Ma', 'Ju', 'Ju', $
      'Au', 'Se', 'Oc', 'No', 'De']
=====
PLOT,x,y1,/nodata,ystyle=4,$ ; untuk nilai chl
      xthick=2., $
      xrange=xrange,$
;      XTICKFORMAT = ['LABEL_DATE','label_date'], $
;      XTICKUNITS = ['months','years'], $
;      XTICKFORMAT = ['label_date'], $
;      XTICKUNITS = ['years'], $
      XTICKinterval= 1, $
      XTickN=Month,$
      xticklen=0.05,$
      charsize=2., $
      charthick=2.2, $
      font = -1
```

- Month = Nantinya akan digunakan untuk mengubah label angka (1 s.d. 12) menjadi seperti yang dituliskan dalam Month
- Plot,x,y1,/nodata..... = digunakan untuk memplotkan data dan pengaturan lainnya. Tetapi karena menggunakan /nodata maka yang digambarkan hanya salah satu sumbunya saja (kasus ini menggunakan pengaturan sumbu X jadi yang digambarkan hanya sumbu X saja).
 - xthick = untuk mengatur ketebalan sumbu
 - xrange = untuk membuat range awal hingga akhir
 - xtickinterval = untuk mengatur interval range
 - xtickn = untuk mengganti label Month tadi
 - xticklen = untuk mengatur ketinggian garis – garis sumbu X
 - charsize = untuk mengatur ukuran karakter
 - chartick = untuk mengatur ketebalan karakter



```
axis,yaxis=0,yrange=yrange,/save, $
YTITLE='($\PI$)', $

ythick=2.2, $
YTICKinterval= 1, $
charsize=2, $
charthick=2.2, $
color = color_array[0], $
font = -1
;stop

oplot,x,y1,$ ;overplot data kedua
color = color_array[0], $
linestyle=12, $
thick=2.0
;stop

oplot,x,y2,$ ; overplot data ketiga
color = color_array[4], $
;linestyle=6, $
thick=2.0
```

- Axis = digunakan untuk membuat sumbu lainnya (Sumbu Y)
- Yaxis = 0, berarti membuat sumbu Y di sebelah kiri (0 kiri, 1 kanan)
- Yrange = untuk range sumbu Y
- Ytitle = memberikan nama sumbu Y
- Color = memberikan warna sumbu Y
- Oplot = untuk memplotkan data grafik
- Linestyle = 0 (berarti solid), 1 (titik titik), 2 dan seterusnya (garis putus – putus)

```
;=====
XYOUTs, 0.5, 0.95, 'Variasi Musiman SST', /normal,alignment=0.5,charsize=2.6, charthick=2.6;, color=color_array[1]
;=====
XYOUTs,0.5, 0.1, 'Month', /normal,alignment=0.5,charsize=2, charthick=2.2, color=color_array[0]
;=====
;=====
file_mkdir,'F:\Praktikum Pemrograman Oseanografi\Hasil\graph'
path_output='F:\Praktikum Pemrograman Oseanografi\Hasil\graph\Grafik.png'
T=TVRD(channel=0,true=1,order=0)
write_png,path_output,T
;=====
```

Seperti pada sebelum – sebelumnya disini merupakan layout data seperti pemberian judul dan keterangan bulan. Setelah itu membuat folder dan menyimpan file png ke dalam folder yang dibuat dengan nama yang telah diatur.



Modul IX. Extract Wind ASCAT

9.1. Extract Wind

Tidak berbeda dengan saat melakukan ekstrak oisst, tujuan dari ekstrak angin adalah mengeluarkan data angin dari file NetCDF menjadi data SAV. Data yang dihasilkan pada pengolahan ini adalah data harian.

```
pro extract_daily_ascat_hires
```

Seperti pada script extract oisst, sebelum memulai scriptnya, harus diawali dengan pro dan judul dari script tersebut.

```
=====
path_list='F:\Interaksi Atmosfer\Data\Wind Ascat\*.nc'
datafiles=findfile(path_list,count=num_files)
;stop
if num_files gt 0 then begin
for x=0,num_files-1 do begin
    pos_filename=strpos(datafiles[x], '_12_')
    sign=strtrim(strmid(datafiles[x],pos_filename+4,3),2)
;stop
```

- Path_list = digunakan untuk meletakkan alamat file dan nama file dari data angin (netcdf).
 - Findfiles = digunakan untuk mencari file yang ada pada path_list dengan kode
 - Strpos (pos_filename) = digunakan untuk mengetahui urutan ke berapa string yang kita inputkan (dalam kasus ini '_12_')
 - Sign = digunakan untuk mendeteksi nama setelah "_12_" atau sesuai dengan string yang kita buat.
-

```
fid=NCDF_OPEN(datafiles[x])
varnames = ncdf_vardir(fid)
;stop
NCDF_VARGET, fid,0,vl0
NCDF_VARGET, fid,1,time
NCDF_VARGET, fid,2,lat
NCDF_VARGET, fid,3,lon
NCDF_VARGET, fid,4,ul0
```

Saat melakukan extract OISST, kita juga menggunakan script seperti ini. NCDF_OPEN digunakan untuk membuka file NetCDF, NCDF_VARDIR digunakan untuk mengetahui variable apa saja yang ada dalam file NetCDF tersebut. NCDF_VARGET digunakan untuk mengambil data variable dari NetCDF untuk disimpan sementara dalam Memory atau RAM



```
lonmin=min(lon)
latmin=min(lat)
grid_interval=(lon[1]-lon[0])
;grid_interval2=(lat[0]-lat[1])
;=====
dimens=size(u10,/dim)
widht=dimens[0]
height=dimens[1]
```

Pada script ini, kita mencari tahu nilai minimal dari longitude dan latitude dengan menggunakan kode “min”. Lalu mencari tahu resolusi data atau grid_interval menggunakan rumus tersebut.

Selanjutnya, untuk mengetahui dimensi data angin menggunakan size(xxxx,/dim). Untuk mengetahui width atau lebarnya menggunakan kode dimens[0] sedangkan untuk mendapatkan height atau tingginya menggunakan kode dimens[1]

```
attnames = ncdf_attdir(fid,'eastward_wind');-
;stop
NCDF_ATTGET,fid,'eastward_wind','scale_factor',scale_factor1
NCDF_ATTGET,fid,'eastward_wind','add_offset',add_offset1
NCDF_ATTGET,fid,'eastward_wind','_FillValue',_FillValue1
NCDF_ATTGET,fid,'eastward_wind','missing_value',missing_value1
NCDF_ATTGET,fid,'eastward_wind','units',units1
NCDF_ATTGET,fid,'eastward_wind','long_name',long_name1
;=====
NCDF_ATTGET,fid,'northward_wind','scale_factor',scale_factor2
NCDF_ATTGET,fid,'northward_wind','add_offset',add_offset2
NCDF_ATTGET,fid,'northward_wind','_FillValue',_FillValue2
NCDF_ATTGET,fid,'northward_wind','missing_value',missing_value2
NCDF_ATTGET,fid,'northward_wind','units',units2
NCDF_ATTGET,fid,'northward_wind','long_name',long_name2
;=====
attnames = ncdf_attdir(fid,'time');-
NCDF_ATTGET,fid,'time','units',units
NCDF_CLOSE,fid
```

Sama seperti saat extract oisst, pada setiap variable yang ada dalam data NetCDF selalu memiliki attribute. Untuk mencari attribute dari data angin dan waktu digunakan kode ncdf_attdir. Selanjutnya untuk menyimpan attribute tersebut ke memory atau RAM menggunakan NCDF_ATTGET. Dan untuk menutup file NetCDF yang telah dibuka menggunakan NCDF_CLOSE.

```
ugd=u10*scale_factor1+add_offset1
vgd=v10*scale_factor2+add_offset2
speed=sqrt(ugd^2+vgd^2)
idx= where(speed > 100, count1)
ugd[idx]=999.
vgd[idx]=999.
;stop
;=====
time_length= size(time,/dim)
;stop
time_length=time_length[0]
count_time=0
;stop
```



Pada script ini, dikarenakan data dalam file NetCDF angin nilainya tidak sebenarnya, jadi harus di konversi terlebih dahulu menggunakan scale_factor dan add_offset.

```
for num_time=0,time_length-1 do begin
juldate=julday(1,1,1990,0,0,time[num_time])
caldat,juldate,month,day,year
print,year,month,day
;stop
year_st=strtrim(string(year,format='(I4.4)'),2)
month_st=strtrim(string(month,format='(I2.2)'),2)
day_st=strtrim(string(day,format='(I2.2)'),2)
```

Jika pada data dari SST merupakan data 2D atau hanya ada data Longitude dan Latitude dalam 1 file, untuk data dari angin, memiliki 1 dimensi tambahan yaitu waktu, sehingga data dari file NetCDF angin merupakan data 3D.

Untuk memecah data tersebut supaya dapat disimpan seperti pada ekstrak oisst, maka digunakan looping kembali.

Julday = digunakan untuk mengkonversi data hari, bulan, tahun, jam, menit, detik ke dalam bentuk Julian.

Julian merupakan bentuk waktu yang dibaca oleh computer. Sedangkan untuk bentuk waktu yang dibaca oleh manusia biasanya disebut dengan Gregorian.

Caldat = digunakan untuk mengkonversi data waktu Julian kedalam bentuk Gregorian.

```
;stop
data_map= ugd[:,*,num_time]
;stop
file_mkdir, 'F:\Interaksi Atmosfer\Olah\ASCAT\ugd\'+year_st+'\' +month_st
save,data_map,filename='F:\Interaksi Atmosfer\Olah\ASCAT\ugd\'+year_st+'\' +month_st+'\' +ugd\'+year_st+month_st+day_st+'_' +sign+
data_map= vgd[:,*,num_time]
file_mkdir, 'F:\Interaksi Atmosfer\Olah\ASCAT\vgd\'+year_st+'\' +month_st
save,data_map,filename='F:\Interaksi Atmosfer\Olah\ASCAT\vgd\'+year_st+'\' +month_st+'\' +vgd\'+year_st+month_st+day_st+'_' +sign+
```

Selanjutnya data angin dipisah perwaktu terlebih dahulu dan disimpan pada variabel data_map. Kemudian data_map disimpan ke dalam bentuk sav menggunakan “save”.



Modul X. Compile Monthly ASCAT

10.1. Compile Monthly

Pada Compile Data Monthly ini berfungsi untuk merata-ratakan data harian menjadi data bulanan. Hal ini biasa dilakukan dalam penelitian dikarenakan data bulanan yang paling mudah untuk dilihat dan dianalisis baik dalam bentuk sebaran spasial maupun dalam bentuk grafik jika data tersebut merupakan data timeseries yang cukup panjang.

```
pro compile_monthly_ascat_hires
=====
TIC
;-----image_coverage-----
latmin=-12
lonmin=91
latmax=12
lonmax=147.
;-----spatial_resolution-----
grid_interval=0.125
;-----data_dimension-----
image_width=round((lonmax-lonmin)/grid_interval)
image_height=round((latmax-latmin)/grid_interval+1)
;stop
=====
```

Annotations in the image show the following values being set or calculated:

- `width = 448` (from `width=dimensions[0]`)
- `grid_interval = 0.12500000`
- `image_width = 448` (from `image_width=round((lonmax-lonmin)/grid_interval)`)
- `height = 193` (from `height=dimensions[1]` and `image_height=round((latmax-latmin)/grid_interval+1)`)

Pro digunakan untuk mengawali script dan dilanjutkan dengan nama scriptnya. Kemudian pada `image_coverage` yang merupakan area yang kita input saat mendownload datanya. `Grid_interval` didapat dari script extract diatas. Kemudian `image_width` dan `image_height` disamakan dengan dimensi data `width` dan `height` pada extract

```
=====
start_year=2017
end_year=2019
for year=start_year,end_year do begin
year_st=strtrim(string(year,format='(I4.4)'),2)
;=====
start_month=1
end_month=12
for month=start_month,end_month do begin
month_st=strtrim(string(month,format='(I2.2)'),2)
;=====
```

Konsepnya sama seperti kompilasi bulanan pada sst. Untuk pengulangannya karena yang dicari adalah bulanan, maka looping juga hanya sampai bulanan.

```
;-----initial_condition_for_calculating_monthly_compilation-----
U_monthly_map=fltarr(image_width,image_height)
U_num_monthly_map=fltarr(image_width,image_height)
v_monthly_map=fltarr(image_width,image_height)
v_num_monthly_map=fltarr(image_width,image_height)
;=====
```

Pada script ini merupakan penjumlahan data angin komponen u dan v baik dari jumlah nilai anginnya maupun jumlah data. Jumlah nilai anginnya berada pada `U_monthly_map` dan `V_monthly_map`. Sedangkan untuk banyaknya data berada pada `U_num_monthly` dan `V_num_monthly`.




```
path_input3='F:\Interaksi Atmosfer\Olah\ASCAT\ugd\' + year_st + \'\' + month_st + \'\' + \'ugd_*.sav'
datanc3=findfile(path_input3,count=num_ncf3)
if num_ncf3 gt 0 then begin
for inc3=0,num_ncf3-1 do begin
;stop
;=====
pos_filename3=strpos(datanc3[inc3],\'ugd_\')
filename3=strtrim(strmid(datanc3[inc3],pos_filename3+5,16),2)
print, filename3
;stop
;=====
restore, datanc3[inc3]
emtx=data_map
restore, 'F:\Interaksi Atmosfer\Olah\ASCAT\vgd\' + year_st + \'\' + month_st + \'\' + \'vgd_' + filename3
emty=data_map
;stop
;=====
```

Open data dari extract wind ascats. Karena data yang disimpan ada 2 jenis yaitu U dan V maka saat pemanggilan data juga harus ada 2 yang dipanggil yaitu data U dan V.

```
indx_u=where(emtx lt 900.,count_u)
indx_v=where(emty lt 900.,count_v)
;stop
;=====default_for_monthly_mean_calculation=====
u_monthly_map[indx_u]=u_monthly_map[indx_u]+emtx[indx_u]
u_num_monthly_map[indx_u]=u_num_monthly_map[indx_u]+1
;stop
v_monthly_map[indx_v]=v_monthly_map[indx_v]+emty[indx_v]
v_num_monthly_map[indx_v]=v_num_monthly_map[indx_v]+1
;stop
;=====
```

Pada pengolahan ini, data angin mulai dijumlahkan dan dihitung banyaknya data.

```
;*****AVERAGING (default)*****
indx_u_monthly=where(u_num_monthly_map gt 0.,count_indx_u_monthly)

monthly_map_u=fltarr(image_width,image_height)+999.
;stop
monthly_map_u[indx_u_monthly]=u_monthly_map[indx_u_monthly]/float(u_num_monthly_map[indx_u_monthly])

indx_v_monthly=where(v_num_monthly_map gt 0.,count_indx_v_monthly)

monthly_map_v=fltarr(image_width,image_height)+999.
;stop
monthly_map_v[indx_v_monthly]=v_monthly_map[indx_v_monthly]/float(v_num_monthly_map[indx_v_monthly])
;=====
```

Setelah melakukan penjumlahan data, maka harus dilakukan pembagian antara jumlah nilai angin dengan banyaknya data angin.

```
;=====
file_mkdir, 'F:\Interaksi Atmosfer\Olah\ASCAT\Monthly\ugd\' + year_st + \'\'
file_mkdir, 'F:\Interaksi Atmosfer\Olah\ASCAT\Monthly\vgd\' + year_st + \'\'
save, monthly_map_u, filename = 'F:\Interaksi Atmosfer\Olah\ASCAT\Monthly\ugd\' + year_st + \'\' + \'ugd_' + year_st + month_st + \'.sav'
save, monthly_map_v, filename = 'F:\Interaksi Atmosfer\Olah\ASCAT\Monthly\vgd\' + year_st + \'\' + \'vgd_' + year_st + month_st + \'.sav'
print, year, month
```

Setelah data dibagi, data siap disimpan. Untuk menyimpannya dilakukan dengan script “save”.



Modul XI. Compile Climatology Wind ASCAT

11.1. Compile Climatology Wind

Setelah melakukan compile monthly, untuk mendapatkan data climatology bulanan maka dilakukan dengan script Compile Climatology. Data yang digunakan pada script ini adalah data dari hasil compile monthly.

```
=====
pro compile_climatology_ascat_hires
=====
TIC
;-----image_coverage-----
latmin=-12
lonmin=91
latmax=12
lonmax=147.
=====
;---spatial_resolution---
grid_interval=0.125
;-----data_dimension-----
image_width=round((lonmax-lonmin)/grid_interval)
image_height=round((latmax-latmin)/grid_interval+1)
;stop
```

Sebelum script dimulai, didahului dengan pro dan nama scriptnya selanjutnya sama seperti sebelumnya mengatur data coverage di latmin, latmax, lonmin, dan lonmax. Resolusi spasial juga disamakan dan dimensi datanya harus sama seperti saat ekstrak, compile monthly, compile climatology.

```
start_month=1
end_month=12
for month=start_month,end_month do begin
month_st=strtrim(string(month,format='(I2.2)'),2)
;=====
;---initial_condition_for_calculating_monthly_compilation---
U_monthly_map=fltarr(image_width,image_height)
U_num_monthly_map=fltarr(image_width,image_height)
v_monthly_map=fltarr(image_width,image_height)
v_num_monthly_map=fltarr(image_width,image_height)
;=====
;=====
start_year=2017
end_year=2019
for year=start_year,end_year do begin
year_st=strtrim(string(year,format='(I4.4)'),2)
;=====
path_input3='F:\Interaksi Atmosfer\Olah\ASCAT\Monthly\ugd\' + year_st + '\ugd_' + year_st + month_st + '.sav'
datanc3=findfile(path_input3,count=num_ncf3)
if num_ncf3 gt 0 then begin
for inc3=0,num_ncf3-1 do begin
;=====
restore, datanc3[inc3]
emtx=monthly_map_u
restore, 'F:\Interaksi Atmosfer\Olah\ASCAT\Monthly\vgd\' + year_st + '\vgd_' + year_st + month_st + '.sav'
emty=monthly_map_v
;=====
```

Untuk langkahnya hampir mirip dengan compile montly bedanya hanya saat melakukan looping yaitu looping bulanan terlebih dahulu kemudian baru looping tahunan. Setelah mengatur loopingnya, maka alamat file dan nama file dimasukkan kedalam path_input yang nantinya akan dibuka dengan menggunakan restorev.

```
;=====default_for_monthly_mean_calculation=====
u_monthly_map[indx_u]=u_monthly_map[indx_u]+emtx[indx_u]
u_num_monthly_map[indx_u]=u_num_monthly_map[indx_u]+1
stop
v_monthly_map[indx_v]=v_monthly_map[indx_v]+emty[indx_v]
v_num_monthly_map[indx_v]=v_num_monthly_map[indx_v]+1
stop
;=====
endfor
endif
endfor
;=====
```

Pada script ini digunakan untuk menjumlahkan data angin komponen u dan v serta menghitung jumlah data angin.

```
;*****AVERAGING (default)*****
indx_u_monthly=where(u_num_monthly_map > 0.,count_indx_u_monthly)

monthly_map_u=fltarr(image_width,image_height)+999.
;stop
monthly_map_u[indx_u_monthly]=u_monthly_map[indx_u_monthly]/float(u_num_monthly_map[indx_u_monthly])
;stop
indx_v_monthly=where(v_num_monthly_map > 0.,count_indx_v_monthly)

monthly_map_v=fltarr(image_width,image_height)+999.
;stop
monthly_map_v[indx_v_monthly]=v_monthly_map[indx_v_monthly]/float(v_num_monthly_map[indx_v_monthly])
;=====
```

Pada script ini digunakan untuk merata-ratakan kecepatan angin komponun u dan v dengan membagi total kecepatan angin (tiap komponen) dengan jumlah data angin (tiap komponen)

```
;=====
file_mkdir,'F:\Interaksi Atmosfer\Olah\ASCAT\Climatology\ugd\
file_mkdir,'F:\Interaksi Atmosfer\Olah\ASCAT\Climatology\vgd\
;=====
data_map=monthly_map_u
save, data_map, filename ='F:\Interaksi Atmosfer\Olah\ASCAT\Climatology\ugd\ugd_'+month_st+'.sav'
data_map=monthly_map_v
save, data_map, filename ='F:\Interaksi Atmosfer\Olah\ASCAT\Climatology\vgd\vgd_'+month_st+'.sav'
;=====
print,month
endfor
TOC
print,'finish'
stop

end
```

Setelah mendapatkan data rata-rata bulanan klimatologi angin tiap komponen, selanjutnya data disimpan kedalam bentuk sav. Hasil dari script ini berupa data 12 bulan saja.



Modul XII. Draw Vector Climatology Wind

12.1. Draw Vector Climatology Wind

Data angin yang telah dilakukan pengolahan Compile Climatology, dapat divisualisasikan menggunakan Draw Vector Climatology. Jika pada Draw Climatology SST kita menggunakan warna pada seluruh lautannya, pada Draw Vector Climatology ini kita akan divisualisasikannya menggunakan bentuk vector dan warna.

```
function true2lon,rgb
;rgb is array of lonarr(3)
return,(long(rgb[0]) or ishft(long(rgb[1]),8) or ishft(long(rgb[2]),16))
end
;=====
pro draw_vector_climatology_ws_ascat_hires
;=====
;-----background_color-----
!p.background='ffffff'x ;---putih---
!p.color=0 ;---hitam---
;=====
;-----window_region-----
;!P.Region=[0.02,0.09,0.98,0.98]
!P.Position=[0.07,0.05,0.95,0.96]
;=====
;---window_pixels---
xsize=1000
ysize=800
;*****
;-----image_coverage-----
latmin=-12
latmax=12
lonmin=91
lonmax=147
;=====
grid_interval =0.125
;=====
width = round((lonmax-lonmin)/grid_interval)
height = round((latmax-latmin)/grid_interval+1)
;=====
```

Masih sama seperti sebelumnya, sebelum melakukan pengolahan data kita harus mengatur kembali lonmin, lonmax, latmin, latmax, grid_interval, width dan height.

Pada bagian window_pixel ada xsize dan ysize yang berfungsi untuk mengatur besaran pixel pada gambar.



```
,
lon=lonmin+findgen(round(widht))*grid_interval
lat=latmin+findgen(round(height))*grid_interval
;stop
;=====
;;-----ploting image
;Indonesia
latmin_cut = -12
latmax_cut = 12
lonmin_cut = 91
lonmax_cut = 147
;=====
;stop

;=====
lat_map=fltarr(widht,height)
;stop
for i=0,widht-1 do begin
lat_map[i,*]=lat
endfor

lon_map=fltarr(widht,height)
;stop
for i=0,height-1 do begin
lon_map[*,i]=lon
endfor
;stop
```

Pada script ini, kita akan mengatur tempat untuk menggambarkan peta kita, untuk mengubahnya dapat diganti pada bagian ploting image.

```
;-----color_table_default-----
loadct,33,/silent
tvlct,r,g,b,/get
rgb1=(long(r) or ishft(long(g),8) or ishft(long(b),16))
rgb2=[[[rgb1 and 'ff'x]],[[ishft(rgb1 and 'ff00'x,-8)]],[[ishft(rgb1 and 'ff0000'x,-16)]]]

bar_length=fix(xsize*0.7)
rgb3=byte(cogrid(rgb2,bar_length,15,3))
;stop
```

Disini kita dapat memilih warna apa yang akan digunakan (pada contoh ini menggunakan kode 33 pada loadct). Untuk list warna dapat dilihat pada link berikut:

<https://www.l3harrisgeospatial.com/docs/loadingdefaultcolortables.html>



```

;=====
start_month=1
end_month=12
for month=start_month,end_month do begin
month_st=strtrim(string(month,format='(I2.2)'),2)
;=====
path_input3='F:\Interaksi Atmosfer\Olah\ASCAT\Climatology\ugd\ugd_'+month_st+'.sav'
;=====
restore,path_input3
map_u=data_map
;stop
;=====
path_input4='F:\Interaksi Atmosfer\Olah\ASCAT\Climatology\vgd\vgd_'+month_st+'.sav'
restore,path_input4
map_v=data_map
;=====
ws = sqrt(map_u^2+map_v^2)
;stop
;=====
window,0,xsize=xsize,ysize=ysize,/,pixmap
;stop
map_set,limit = [latmin_cut, lonmin_cut, latmax_cut, lonmax_cut],/iso
;stop
;=====
ws_map=ws
;=====
ws_map2 = ws_map;[a:b,c:d]
ws_map3 = ws_map2
;=====

```

Karena menggunakan data monthly climatology, maka looping yang digunakan hanyalah looping bulanan saja. Untuk input file yang digunakan merupakan input file hasil dari compile climatology.

```

;=====
ws_map=ws
;=====
ws_map2 = ws_map;[a:b,c:d]
ws_map3 = ws_map2
;=====
idx = where(ws_map gt 0.01 and ws_map lt 10,countt)
speed=mean(ws_map[idx])
;stop
;=====
max_value=10.
min_value=0.
;=====
idx_low = where (ws_map2 le min_value)
idx_high = where (ws_map2 ge max_value and ws_map2 lt 200)
;stop
;=====
ws_map2[idx_low] = min_value
ws_map2[idx_high] = max_value
;=====

```

Pada script ini kita mengatur nilai maksimal dan minimal anginnya. Untuk mengaturnya dapat digunakan pada variable max_value dan min_value.

```

;=====
level_interval = 0.01
;=====
;-----default_4_contour_level-----
numlevel=fix((max_value-min_value)/level_interval+1)
col=lonarr(numlevel)
;stop
for c=0,numlevel-1 do begin
cc=byte(c*level_interval/(max_value-min_value)*255)
col[c]=true2lon(rgb2[cc,0,*])
endfor
;stop
;=====
;-----array_4_contour_interval-----
levels=findgen(numlevel)*level_interval+min_value
;stop
;-----contour interval-----

```

Pada script tersebut. Warna yang dibutuhkan untuk mevisualisasikan data dibuat.



```
;-----color_legend_&_position-----
tv,rgb3, tr=3,0.15,0.12, /normal
;stop
;-----interval_value_legend-----
num_interval=5.0
;=====
;-----loop_legend-----
for w=0,num_interval do begin

index_value=w*(max_value-min_value)/num_interval+min_value
index_plot=string(index_value,format='(F6.1)')
x_position=0.15+0.7/num_interval*w
y_position=0.08
xyouts,x_position,y_position,index_plot, charsize=2.5, charthick=2.5, font=1,/normal,$
alignment=0.5

endfor
;stop
;=====
ws_unit=' (m/s) '
XYOUTs,0.9,0.08,ws_unit,/normal,alignment=0.1,charsize=2.5,charthick=2,font=1
;stop
```

Untuk memberikan keterangan dilakukan pada script tersebut. Tv,rgb3 digunakan untuk memberikan keterangan warna. Selanjutnya pada bagian loop legend digunakan untuk memberikan keterangan angka pada bar warna. Dan terakhir ada ws_units dan XYOUTs digunakan untuk memberikan keterangan satuan dari data.

```
sampling=8
width2=round(width/sampling)
height2=round(height/sampling)
map_u=congrid(map_u,width2,height2)
map_v=congrid(map_v,width2,height2)
lon_map=congrid(lon_map,width2,height2)
lat_map=congrid(lat_map,width2,height2)

u_array=reform(map_u,width2*height2)
v_array=reform(map_v,width2*height2)
lat_array=reform(lat_map,width2*height2)
lon_array=reform(lon_map,width2*height2)
;stop

in= where(u_array lt 50, count_in)
if count_in ne 0 then begin
u_array= u_array[in]
v_array= v_array[in]
lat_array= lat_array[in]
lon_array= lon_array[in]
endif
;stop
;=====
index=where((lat_array ge latmin_cut)and(lat_array le latmax_cut)and $
(lon_array ge lonmin_cut)and(lon_array le lonmax_cut),count)

u_array=u_array[index]
v_array=v_array[index]
lat_array=lat_array[index]
lon_array=lon_array[index]

my_varrows,u_array,v_array,lon_array,lat_array,rgb2, Length =2 ,Title = title, $
Color='666666'x,Thick=1.5,mag_ref=max_value, $
min_value=min_value;,/monotone
```

Untuk menggambarkan vector dan warna pada peta, menggunakan script tersebut. Disini terdapat variable sampling digunakan untuk mengatur kerapatan vector angin. Semakin kecil sampling yang digunakan maka kerapatan anginnyapun semakin kecil.

My_varrows, merupakan script yang digunakan untuk memplot vector angin. Untuk mengatur panjang pendeknya arah angin dapat menggunakan length (pada my_varrows) dan untuk mengatur ketebalan arah anginnya dapat menggunakan thick.




```
;=====
XYOUTS,0.5, 0.9, 'Wind '+month_st, /normal,alignment=0.5,charsize=3, charthick=2.5
XYOUTS,0.5,0.16,'Longitude (!U0!NE)',/normal,alignment=0.5,charsize=2.5,charthick=2,font=1, color='000000'x1
XYOUTS,0.03,0.5,'Latitude (!U0!NN)',/normal,alignment=0.5,charsize=2.5,charthick=2,font=1, orientation=90, color='000000'x1
;=====
```

Selanjutnya pada script ini, kita akan memberikan Judul, keterangan longitude dan latitude pada Peta.

```
;stop
file_mkdir, 'F:\Interaksi Atmosfer\Hasil\ascat\Climatology'
path_output='F:\Interaksi Atmosfer\Hasil\ascat\Climatology\wind'+month_st+'.png'
T=TVRD(channel=0,true=1,order=0)
write_png,path_output,T
;=====
endfor
print,'finish'
;=====
!P.Region=0
!P.Position=0
;=====
end

;-----program_is_done!-----
```

Setelah semua siap. Untuk menyimpan data gambar angin dapat menggunakan script berikut.
Dimana :

File_mkdir = digunakan untuk membuat folder

Path_output = meletakkan alamat folder dan nama file

Write_png = melukiskan gambar dari IDL ke bentuk png



Modul XIII. Convert Climatology Ascst Timeseries

13.1. Convert Climatology Ascst Timeseries

Script ini digunakan untuk mengkonversi data climatology (12 file) menjadi data timeseries (1 file) yang berisi 12 bulan.

```
=====
> pro convert_climatology_ascst_timeseries
=====
TIC
;-----image_coverage-----
latmin=-12.
lonmin=91.
latmax=12.
lonmax=147.
> =====
;---spatial_resolution---
grid_interval=0.125
;-----data_dimension-----
image_width=round((lonmax-lonmin)/grid_interval)
image_height=round((latmax-latmin)/grid_interval+1)
count_msst=0
> stop
;*****
;-----loop_for_month-----
start_month=1
end_month=12
for month=start_month,end_month do begin
month_st=strtrim(string(month,format='(I2.2)'),2)
=====
```

Seperti sebelum – sebelumnya script dimulai dengan pro, lalu data coverage diatur sesuai dengan wilayah saat cropping, grid_interval yang disamakan juga dan data dimension juga disamakan.

```
-----loop_for_month-----90_ID
start_month=1
end_month=12
for month=start_month,end_month do begin
month_st=strtrim(string(month,format='(I2.2)'),2)
=====
path_input3='F:\Interaksi Atmosfer\Olah\ASCAT\Climatology\ugd\ugd_'+month_st+'.sav'
restore,path_input3
u_map=data_map
restore,'F:\Interaksi Atmosfer\Olah\ASCAT\Climatology\vgd\vgd_'+month_st+'.sav'
v_map=data_map
ws=sqrt(u_map^2+v_map^2)
```

Sama seperti draw, karena menggunakan data climatology maka looping hanya bulan saja. Alamat file dan nama file diletakkan pada path_input3 dan dibuka dengan restore. Data dipanggil menggunakan data_map karena saat compile climatology datanya disimpan dengan data_map



```
;=====case1 natuna
nama1='natuna'
latmin_cut1=5
lonmin_cut1=107
latmax_cut1=8
lonmax_cut1=110
;=====case2 selatan jawa
nama2='seljaw'
latmin_cut2=5
lonmin_cut2=107
latmax_cut2=8
lonmax_cut2=110
;=====
a1 = fix ((lonmin_cut1-lonmin)/grid_interval)
b1 = fix ((lonmax_cut1-lonmin)/grid_interval)
c1 = fix ((latmin_cut1-latmin)/grid_interval)
d1 = fix ((latmax_cut1-latmin)/grid_interval)
;stop
ws21= ws[a1:b1,c1:d1]

a2 = fix ((lonmin_cut2-lonmin)/grid_interval)
b2 = fix ((lonmax_cut2-lonmin)/grid_interval)
c2 = fix ((latmin_cut2-latmin)/grid_interval)
d2 = fix ((latmax_cut2-latmin)/grid_interval)
;stop
ws22= ws[a2:b2,c2:d2]
```

Seperti saat melakukan cropping pada ekstrak, di script ini juga melakukan cropping data untuk menentukan lokasi yang akan dibuat grafiknya. Script ini mengambil 2 wilayah di case1 dan case2. Sebenarnya disini tidak harus 2 wilayah dapat berupa 1 wilayah, 2 wilayah, 3 wilayah dan seterusnya.

```
;===== 1
;stop
indx_ws1=where(ws21 lt 999.0)
ws_mean1=mean(ws21[indx_ws1])
;stop
;===== 2
indx_ws2=where(ws22 lt 999.0)
ws_mean2=mean(ws22[indx_ws2])
;=====
```

Data yang sudah di cropping pada script diatas akan diubah menjadi 1 data dengan menggunakan rata-rata (mean). Sebelumnya dilakukan filtering terlebih dahulu untuk nilai error agar nilai error tidak masuk perhitungan rata-ratanya.

```
;stop
if(count_msst eq 0) then begin
dataset1=[ws_mean1,ws_mean2,month]
;stop
endif else begin
dataset1=[[dataset1],[ws_mean1,ws_mean2,month]]
;stop
endelse
;stop
count_msst=count_msst+1
;stop
endfor
```

Pada script ini, data perbulan tadi diubah menjadi data timeseries. Dimana urutan kolomnya harus disesuaikan dengan data yang diconvert ke dalam timeseries. Pada script convert ini menggunakan 2 wilayah (case1 dan case2) dan juga waktu (bulan) sehingga menggunakan 3 kolom untuk datanya (SST case1, SST case2 dan bulan).



```
=====
file_mkdir, 'F:\Interaksi Atmosfer\Olah\Timeseries\Ascat'
=====
path_output1='F:\Interaksi Atmosfer\Olah\Timeseries\Ascat\timeseries.txt'
format1='(3F15.5)'
openw,3,path_output1
printf,3,dataset1, format=format1
close,3
=====
TOC
print, 'finish'
stop
end
```

- g. File_mkdir = membuat folder
 - h. Path_output = alamat penyimpanan dan nama file
 - i. Format1 = merupakan format yang akan digunakan untuk penulisan pada file txt. (3F15.5) artinya, 3 = 3 kolom yang digunakan tadi, F merupakan Float (Bilangan decimal), 15 merupakan 15 karakter yang boleh dituliskan pada text nanti, .5 = 5 angka di belakang koma
 - j. Openw = untuk membuka file txt / membuat file txt dan menuliskan file kedalamnya (open and write)
 - k. Printf = untuk menuliskan data ke dalam txt tadi. Yang dituliskan berupa variable dataset dengan format yaitu format1
 - l. Close = untuk menutup file yang tadi dibuka di openw (supaya filenya tidak menyangkut di computer)
-



Modul XIV. Plot Timeseries SST and Wind

14.1. Plot Timeseries OISST and Ascat

Setelah melakukan convert tadi, maka data yang telah jadi timeseries dapat digambarkan menjadi grafik menggunakan script `plot_time_series_oisst_ascat.pro`.

```
=====
pro plot_time_series_oisst_ascat
=====
;-----background_color-----
!p.background='ffffff'x ;---putih---
!p.color=0 ;---hitam---
;-----window_region-----
!P.Region=[0.01,0.09,0.85,0.94]
;-----window_pixel-----
xsize=800
ysize=800
=====
```

Seperti biasa, dimulai dengan `pro` dan nama scriptnya. Dan mirip saat `draw`, fungsi `background` untuk memberikan warna background pada canvas, `color` untuk memberikan warna tulisan, `region` untuk memberikan layout. Dan `window pixel` untuk memberikan ukuran pixel canvasnya.

```
=====input_data=====
path_input_all = 'F:\Praktikum Pemrograman Oseanografi\Pengolahan\Timeseries\timeseries.txt'
PRINT, 'path_input_all: ', path_input_all
;-----reading_ascii_data (default)-----
;stop
openr, 1, path_input_all
Each_record_sst_all = ftrarr (3) ;====number of columns====
count_sst_all = 0
; stop
WHILE NOT EOF (1) DO BEGIN
  READF, 1, Each_record_sst_all
;stop
  if (count_sst_all eq 0) then begin
    Record_sst_all = Each_record_sst_all
    count_sst_all = count_sst_all+1
  ; stop
  endif else begin
    Record_sst_all = [[Record_sst_all], [Each_record_sst_all]]
  ; stop
  endwhile
;stop
ENDWHILE
;stop
close, 1
```

- g. `Path_input_all` = untuk meletakkan alamat dan nama file txt hasil convert tadi.
- h. `Openr` = digunakan untuk membuka dan membaca file txt (open and read), diikuti dengan alamat dan nama file di `path_input_all`
- i. Sebelum masuk lebih lanjut, perlu mengatur jumlah kolom yang digunakan saat convert, karena menggunakan 3 kolom maka pada `fltarr(x)` ditulis dengan 3 ($x = 3$) menjadi `fltarr(3)`
- j. `While not EOF (1) do begin` = digunakan untuk melooping data hingga datanya habis
- k. `Readf` = membaca data per baris. Data perbaris akan dibuat menjadi seperti dataset di script convert
- l. `Close` = file txt yang di open di tutup.

```
*****
sst_all = size(Record_sst_all, /dimension)
num_data_sst_all = sst_all[1]
kolom=sst_all[0]
baris=sst_all[1]

print, num_data_sst_all
```

Pada script ini, kita hanya mengidentifikasi jumlah kolom dan baris dari data OISST

```
data_wind=read_table('F:\Interaksi Atmosfer\Olah\Timeseries\Ascat\timeseries.txt')
```

Mirip seperti cara membuka data txt OISST, hanya saja bentuk lain dari membuka file text dapat menggunakan read_table.

```
=====dataset_array=====
sst1 = transpose(record_sst_all[0,:]);Natuna
sst2 = reform (record_sst_all[1,:], num_data_sst_all);Selatan Jawa
ws1 = transpose(data_wind[0,:])
ws2 = transpose(data_wind[1,:])
x = reform (record_sst_all[2,:], num_data_sst_all)
```

Disini kita mengambil data sst dan angin di wilayah 1 dan 2 serta data waktu yang dimasukkan pada variable “x”.

Pada script ini ada kode “transpose” dan “reform” yang fungsinya mirip. Hanya saja untuk transpose tidak bisa digunakan untuk data 3 dimensi, sedangkan reform bisa digunakan untuk data 3 dimensi.

```
=====y_axis_value=====
xrange=[1,12]
sstrange=[25,31]
wsrange=[0,10]
;stop
```

Untuk mengatur range sumbu X dan sumbu Y dapat digunakan pada bagian tersebut.

```
=====
date_label=label_date(date_format=['%M','%Y'])
color_array=['000000'x, 'ff0000'x, '00ff00'x, '0099ff'x, '0000ff'x, '990000'x, '009966'x, '7f7f7f'x, 'db70db'x, 'ffa300'x]
;=====
window,0,xsize=xsize,ysize=ysize
;stop
```

- Label_date = digunakan untuk membuat label berupa data waktu, dalam kasus ini membuat label dalam bulan dan tahun. (umumnya script ini digunakan untuk data timeseries yang panjang seperti data timeseries selama 3 tahun (bukan klimatologi))
- Color_array = berisi kumpulan kode warna untuk plot grafik nanti
- Window = seperti sebelumnya untuk membuat canvas dengan ukuran xsize dan ysize.

```
PLOT,x,sst1,/nodata,ystyle=4,$
xthick=2., $
xrange=xrange,$
; XTICKFORMAT = ['LABEL_DATE','label_date'], $
; XTICKUNITS = ['months','years'], $
; XTICKFORMAT = ['label_date'], $
; XTICKUNITS = ['years'], $
XTICKinterval= 1, $
xticklen=0.05,$
charsize=2., $
charthick=2.2, $
font = -1
```

- Month = Nantinya akan digunakan untuk mengubah label angka (1 s.d. 12) menjadi seperti yang dituliskan dalam Month
- Plot,x,y1,/nodata..... = digunakan untuk memplotkan data dan pengaturan lainnya. Tetapi karena menggunakan /nodata maka yang digambarkan hanya salah satu sumbunya



saja (kasus ini menggunakan pengaturan sumbu X jadi yang digambarkan hanya sumbu X saja).

- b.1. `xthick` = untuk mengatur ketebalan sumbu
 - b.2. `xrange` = untuk membuat range awal hingga akhir
 - b.3. `xtickinterval` = untuk mengatur interval range
 - b.4. `xtickn` = untuk mengganti label Month tadi
 - b.5. `xticklen` = untuk mengatur ketinggian garis – garis sumbu X
 - b.6. `charsize` = untuk mengatur ukuran karakter
 - b.7. `chartick` = untuk mengatur ketebalan karakter
-

```
.  
axis,yaxis=0,yrange=sstrange,/save,$  
YTITLE='Sea Surface Temperature (!U0!NC)',  
ythick=2.2,$  
YTICKinterval= 1,$  
charsize=2,$  
charthick=2.2,$  
color = color_array[0],$  
font = -1  
;stop  
oplot,x,sst1,$  
color = color_array[0],$  
linestyle=0,$  
thick=2.0  
;stop  
; oplot,x,sst2,$  
; color = color_array[3],$  
; ;linestyle=6,$  
; thick=2.0  
;stop
```

- a. `Axis` = digunakan untuk membuat sumbu lainnya (Sumbu Y)
 - b. `Yaxis = 0`, berarti membuat sumbu Y di sebelah kiri (0 kiri, 1 kanan)
 - c. `Yrange` = untuk range sumbu Y
 - d. `Ytitle` = memberikan nama sumbu Y
 - e. `Color` = memberikan warna sumbu Y
 - f. `Oplot` = untuk memplotkan data grafik
 - g. `Linestyle = 0` (berarti solid), 1 (titik titik), 2 dan seterusnya (garis putus – putus)
-



```
axis,yaxis=1,yrange=wsrange,/save, $
YTITLE='Wind Speed (m/s)', $
ythick=2.2, $
YTICKinterval= 1, $
charsize=2, $
charthick=2.2, $
color = color_array[0], $
font = -1
;stop
oplot,x,ws1,$
color = color_array[0], $
linestyle=5,$
thick=2.0

; oplot,x,ws2,$
; color = color_array[3], $
; linestyle=5,$
; thick=2.0
; ;stop
```

Sama seperti bagian SST (diatas), hanya saja Sumbu Y bagian kanan digunakan untuk data angin.

```
XYOUTs, 0.5, 0.95, 'Variasi Musiman SST dan Angin', /normal,alignment=0.5,charsize=2.6, charthick=2.6;, color=color_array[1]
;=====
XYOUTs,0.5, 0.1, 'Month', /normal,alignment=0.5,charsize=2, charthick=2.2, color=color_array[0]
;=====
stop
;*****
;=====
file_mkdir,'F:\Interaksi Atmosfer\Hasil\graph'
path_output='F:\Interaksi Atmosfer\Hasil\graph\Grafik SST Wind.png'
;=====
T=TVRD(channel=0,true=1,order=0)
write_png,path_output,T
;=====
```

Seperti pada sebelum – sebelumnya disini merupakan layout data seperti pemberian judul dan keterangan bulan. Setelah itu membuat folder dan menyimpan file png ke dalam folder yang dibuat dengan nama yang telah diatur.



Modul XV. Tugas

15.1. Tugas

Tugas untuk praktikum ini, adalah menganalisis hasil – hasil yang telah didapat dari draw climatology, draw climatology vector wind ascet dan plot timeseries sst and wind pada wilayah tertentu (misalnya selatan jawa, laut jawa, laut banda dan sebagainya)



Pre Test dan Post Test

Modul 1. Download Data

Pre-Test

1. Data apakah yang digunakan pada Praktikum ini?
2. Apa saja aplikasi yang digunakan pada praktikum ini?
3. Apa nama modul 1 kali ini?

Post-Test

1. Apakah aplikasi yang digunakan untuk mengunduh data SST?
2. Dimanakah kita dapat mendownload data OISST?
3. Apakah yang disebut dengan FTP?

Modul 2. GUI IDL

Pre-Test

1. Apakah judul modul 2 ini?
2. Bahasa apakah yang akan digunakan pada praktikum kali ini?

Post-Test

1. Apasaja tools yang ada pada IDL?
2. Dimanakah kita dapat menulis kode IDL?
3. Tombol apa yang digunakan untuk menjalankan program IDL?

Modul 3. Ekstrak Data

Pre-Test

1. Apakah judul modul 3 ini?
2. Berapakah batas longitude dan latitude Indonesia (minimal dan maksimal)?

Post-Test

1. Kode apa yang digunakan untuk mengawali script yang akan dibuat?
2. Kode apa yang digunakan untuk membuka data NetCDF?
3. Kode apa yang digunakan untuk mengakhiri program?

Modul 4. Compile Data Monthly

Pre-Test

1. Apakah judul modul 4 ini?
2. Apa ekstensi dari script IDL?
3. Apa ekstensi dari file yang disimpan saat pengolahan IDL?

Post-Test

1. Rumus apa yang digunakan pada saat melakukan compile data?



2. Kode apa yang digunakan untuk membuka file yang telah disimpan dalam pengolahan modul 3?
3. Kode apa yang digunakan untuk menyimpan data yang telah diolah?

Modul 5. Compile Data Climatology

Pre-Test

1. Apakah judul modul 5 ini?
2. Berapa jumlah data yang disimpan pada pengolahan sebelumnya dalam 1 tahun?
3. Dalam melakukan looping, kode apa yang digunakan untuk menjalankannya?

Post-Test

1. Apa perbedaan antara Modul Compile Data Monthly dan Compile Data Climatology?
2. Kode apa yang digunakan untuk mengakhiri iterasi (if)?
3. Berapa jumlah data yang dihasilkan pada pengolahan modul ini?

Modul 6. Draw Climatology OISST

Pre-Test

1. Apakah judul modul 6 ini?
2. Apa yang disebut dengan dimensi data?

Post-Test

1. Apa hasil dari modul ini?
2. Apa inputan dari modul ini?
3. Kode warna apa yang digunakan dalam modul ini?

Modul 7. Convert Climatology to Timeseries

Pre-Test

1. Apakah judul modul 7 ini?
2. Apa fungsi modul ini?

Post-Test

1. Apa input yang digunakan dalam modul ini?
2. File apa yang dihasilkan dari modul ini?

Modul 8. Plot Timeseries OISST

Pre-Test

1. Apakah judul modul 8 ini?
2. Apa fungsi modul ini?

Post-Test

1. Apa input yang digunakan dalam modul ini?
2. Apa hasil yang didapat pada modul ini?
3. Tuliskan script terakhir pada modul ini?



Modul 9. Extract Wind Ascet

Pre-Test

1. Apakah judul modul 9 ini?
2. Data angin dari mana yang digunakan untuk praktikum ini?

Post-Test

1. Apa input yang digunakan dalam modul ini?
2. Apa hasil yang didapat pada modul ini?

Modul 10. Compile Monthly Ascet

Pre-Test

1. Apakah judul modul 10 ini?
2. Berapa resolusi dari data Ascet?

Post-Test

1. Apa input yang digunakan dalam modul ini?
2. Apa fungsi dari restore?
3. Apa fungsi dari file_mkdir?

Modul 11. Compile Climatology Wind Ascet

Pre-Test

1. Apakah judul modul 11 ini?
2. Sebutkan dimensi yang ada pada modul ini?

Post-Test

1. Apa input yang digunakan dalam modul ini?
2. Apa fungsi dari for?
3. Apa fungsi dari where?

Modul 12. Draw Vector Climatology Wind

Pre-Test

1. Apakah judul modul 12 ini?
2. Pada saat apa endfor digunakan?

Post-Test

1. Apa hasil dari modul ini?
2. Kode apa yang digunakan untuk menggambarkan vector pada peta?
3. Kode apa yang digunakan untuk membuat keterangan judul, longitude dan latitude pada peta?

Modul 13. Convert Climatology Ascet Timeseries

Pre-Test

1. Apakah judul modul 13 ini?
2. Bagaimana rumus kecepatan angin total bila diketahui kecepatan angin u dan kecepatan angin v?





Post-Test

1. Data Wind Ascat dapat diunduh dari mana?
2. Tuliskan 2 jenis data Ascat yang digunakan?

Modul 14. Plot Timeseries SST and Wind

Pre-Test

1. Apakah judul modul 14 ini?
2. Pada range berapa SST di perairan Indonesia?

Post-Test

1. Apa yang dihasilkan dari modul ini?
2. Bagaimana hubungan dari SST dan Angin?



Responsi

1. Jelaskan urutan yang dilakukan dalam praktikum ini? (Dari download data OISST hingga Plot Timeseries SST and Wind)
2. Sebutkan data apa saja yang digunakan dalam praktikum ini?
3. Sebutkan resolusi temporal dari tiap data tersebut?
4. Apa hubungan yang terlihat dari SST dan Angin?
5. Jelaskan hasil yang kalian dapatkan saat melakukan draw climatology dan plot timeseries (SST dan Angin)?

