

# **Module Praktikum Pemodelan Iklim**

Agung Baruna Setiawan Noor

1/10/23



# Daftar Isi

<b>Pendahuluan</b>	<b>1</b>
<b>1 <i>Dynamical Downscaling: Model Weather Research Forecasting (WRF)</i></b>	<b>3</b>
1.1 Pendahuluan	3
1.2 Dasar-dasar Pemrograman Bash	8
1.3 Prosedur Praktikum: Instalasi WRF	11
1.3.1 Menjalankan <i>Windows Subsystem Linux</i> (WSL) di Windows 10/11	11
1.3.2 Memasang NCAR Command Language (NCL)	12
1.3.3 Sebelum Mulai	13
1.3.4 Data dan Package Pendukung	14
1.3.5 Instalasi Software Compiler	16
1.3.6 Instalasi Package	17
1.3.7 Instalasi WRF	22
1.3.8 Instalasi WRF Pre-Processing (WPS)	24
1.4 Prosedur Praktikum: Menjalankan Simulasi WRF-ARW	25
1.4.1 Program WPS	26
1.4.2 Program WRF	32
1.5 Prosedur Praktikum: Analisis Luaran WRF	40
1.5.1 Membuka Jupyter Notebook	41
1.5.2 Menggunakan <code>wrf-python</code>	41



# Pendahuluan

Tentang pemodelan iklim



# 1 *Dynamical Downscaling*: Model Weather Research Forecasting (WRF)

## 1.1 Pendahuluan

Global Climate Model (GCM) adalah alat utama dan paling komprehensif yang digunakan dalam mensimulasikan keadaan iklim pada masa lalu, masa sekarang, maupun masa depan. GCM memiliki kemampuan dalam melakukan simulasi variabilitas iklim, sifat-sifat fisis, serta kimia di bumi dengan perhitungan secara matematis yang menggambarkan proses, interaksi, dan timbal balik pada atmosfer, lautan, dan biotik. Kelemahan dalam GCM adalah ketidakmampuan menangkap kejadian-kejadian iklim pada skala regional maupun lokal karena memiliki resolusi spasial yang kasar, sekitar  $>100$  km. Ketidakmampuan GCM dalam menjelaskan keadaan iklim secara regional maupun lokal karena keterbatasan sumber daya komputer. Akan tetapi pada saat ini, teknologi komputasi semakin berkembang dan beberapa instansi yang bergerak dalam bidang cuaca dan iklim, seperti Met Office Hadley Center, National Center for Atmospheric Research (NCAR), dan European Centre for Medium-Range Weather Forecasts (ECMWF) telah mengembangkan GCM dalam menjelaskan fenomena-fenomena cuaca dalam skala regional dengan sumber daya komputer yang sangat tinggi.

Metode *downscaling* di dalam ilmu iklim umum digunakan pada kajian hidrologi [8], pertanian [1], dan sistem perkotaan [13]. Sebagai contoh pada kajian hidrologi, teknik *downscaling* digunakan untuk pemodelan debit sungai dan banjir [7]. Metode *downscaling* adalah cara mendapatkan informasi lebih spesifik dengan resolusi tinggi. Kata *downscaling* biasanya merujuk pada peningkatan resolusi spasial maupun temporal. Terdapat dua cara dalam melakukan metode ini, yaitu statistik dan dinamik. Di dalam ilmu iklim, metode ini umum digunakan pada GCM. Wilby dan Wigley [15] mengelompokkan teknik *downscaling* menjadi 4 kategori, yaitu

### 1. Regresi

Metode regresi merupakan metode *downscaling* paling awal yang telah digunakan pada kajian perubahan iklim. Hal ini dapat dibuktikan dari penelitian oleh Kim pada tahun 1984 [4]. Pendekatan ini secara umum membangun hubungan linier atau non-linier antara parameter titik lokasi dengan prediktor variabel dari resolusi kasar. Contoh dari metode ini adalah regresi linier sederhana, regresi linier berganda, Artificial Neural Network (ANN), regresi komponen utama (Principle Component Regression/PCR), dan lain sebagainya. Sudah banyak penelitian yang menerapkan metode ini untuk kajian perubahan iklim [6, 3, 2, 11].

### 2. Pola cuaca

Metode ini dibangun dari hubungan statistik dari variabel cuaca di stasiun observasi atau rata-rata area dengan klasifikasi cuaca tertentu yang dapat diturunkan secara obyektif maupun subyektif. Metode ini secara obyektif dapat meliputi komponen utama, Canonical Correlation Analyses (CCA),

## 1 Dynamical Downscaling: *Model Weather Research Forecasting (WRF)*

aturan Fuzzy, dan Neural Networks. Contoh prosedur pengelompokkan pola cuaca, yaitu European Grosswetterlagen dan British Isles Lamb Weather Types.

### 3. Stokastik

Model WGEN [10] merupakan contoh dari pendekatan ini. Model ini memiliki kemampuan dalam membangkitkan data curah hujan harian berdasarkan peluang kejadian hujan (hujan dan tidak hujan) dengan rantai Markov order satu. Model ini telah digunakan dalam kajian perubahan iklim dan analisis dampak. Model stokastik yang telah diperoleh dari data observasi deret waktu dapat divalidasi dengan GCM dan perlu dikalibrasi terlebih dahulu.

### 4. *Limited-area climate models (LAM)*

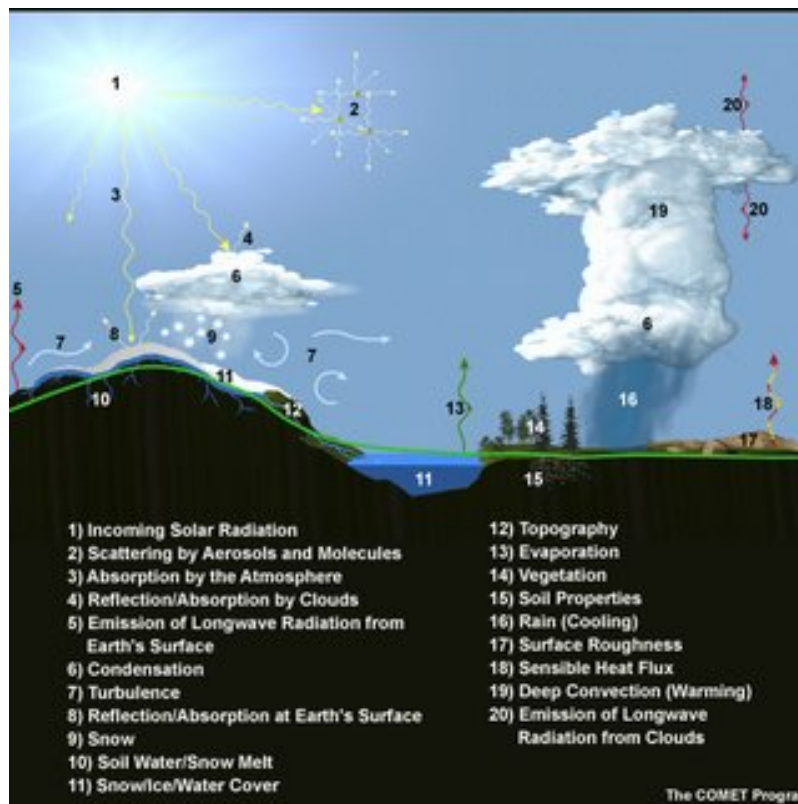
Metode terakhir untuk melakukan *downscaling* terhadap GCM adalah dengan menyematkan model iklim dengan area terbatas. Sebenarnya, LAM pada saat ini dapat diistilahkan sebagai *Regional Climate Model (RCM)*. RCM memiliki resolusi spasial kurang dari 100 km. RCM memerlukan sumber daya komputer yang hampir sama dengan menjalankan GCM. RCM memiliki kemampuan dalam mensimulasikan proses-proses atmosfer pada skala menengah, seperti curah hujan orografis dan konveksi awan atau skala tinggi, seperti turbulensi. Contoh dari RCM adalah Weather Research Forecasting (WRF) [12].

Teknik *dynamical downscaling* dapat menjelaskan proses-proses fisika (hukum termodinamika, hukum kekekalan energi, hukum gerak) untuk setiap skala piksel. Teknik ini membutuhkan kondisi batas menggunakan data GCM dan perlu menentukan pemilihan lokasi. Teknik ini merujuk pada penggunaan RCM untuk meningkatkan skala spasial dan temporal. Model iklim regional meliputi komponen dinamik, fisik, maupun kimia. Komponen dinamik atmosfer memperhitungkan komponen fisika atmosfer yang mencakup proses-proses fisik, seperti radiasi gelombang panjang dan pendek, presipitasi, dan proses pertukaran energi di permukaan bumi atau laut. Proses pada skala sub-piksel dimodelkan oleh berbagai skema parameterisasi yang tentunya dapat berasal dari perhitungan matematika (integral atau diferensial) serta statistik.

WRF merupakan salah satu teknik *dynamical downscaling*. Penggunaan model WRF semakin berkembang saat ini dengan tujuan riset dan operasional. WRF memiliki spesifikasi dalam berbagai aplikasi prediksi di dalam sistem bumi, seperti kimia atmosfer, hidrologi, kebakaran hutan, siklon, dan iklim regional. Sebanyak lebih dari 36.000 pengguna WRF tersebar di 162 negara, termasuk Indonesia yang telah menggunakan WRF untuk kebutuhan operasional [9]. Ada dua jenis model WRF berdasarkan penyelesaian persamaan aliran fluida atmosfer, yaitu *Advanced Research WRF (ARW)* dan *Nonhydrostatic Mesoscale Model (NMM)*. WRF-ARW dikembangkan oleh National Centre of Atmosphere Research (NCAR), sedangkan WRF-NMM dikembangkan oleh National Centre of Environmental Prediction (NCEP) [12]. Secara umum, WRF mampu melakukan simulasi beberapa komponen *Numerical Weather Prediction (NWP)* Gambar 1.1. Berdasarkan aplikasi tertentu, WRF memiliki beberapa turunan model, seperti WRF-Chem (kimia atmosfer), WRF-Hydro (hidrologi), dan WRF-Fire (kebakaran hutan dan lahan). Di dalam praktikum ini, Anda hanya perlu memahami cara instalasi dan simulasi WRF-ARW.

WRF diimplementasikan ke dalam bahasa pemrograman komputer. Jika Anda melihat repositori github WRF, Jenis bahasa pemrograman yang paling banyak digunakan adalah Fortran. Untuk perhitungan fisik, kimia, dan matematis diketik dalam bahasa Fortran. Model WRF dapat dijalankan pada personal komputer hingga super komputer. Sebenarnya, Anda dapat menjalankan WRF dengan 1 prosesor saja. Akan tetapi, waktu proses simulasi mungkin akan lama akibat dari resolusi spasial dan temporal

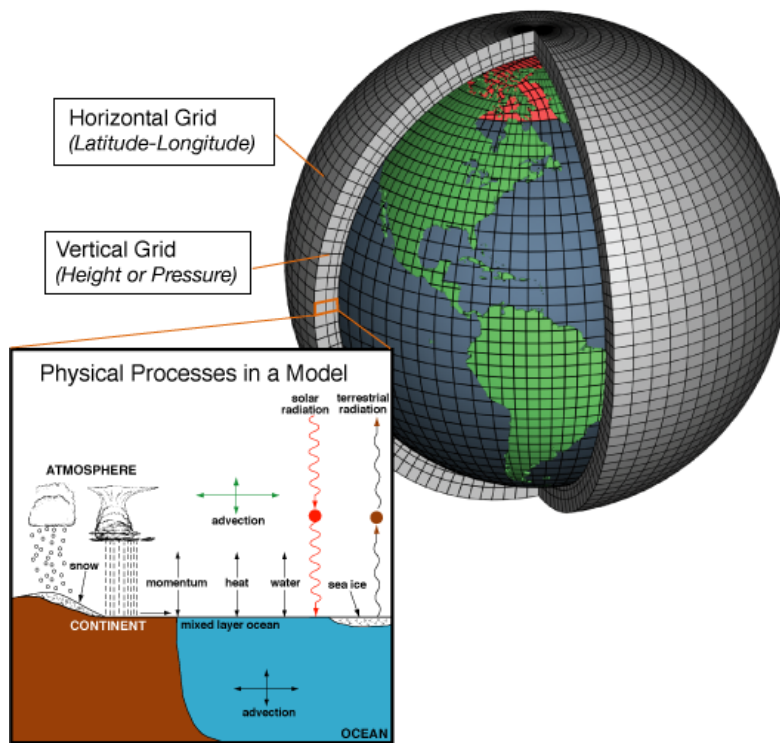




Gambar 1.1: Beberapa parameter dalam *Numerical Weather Prediction* [14]

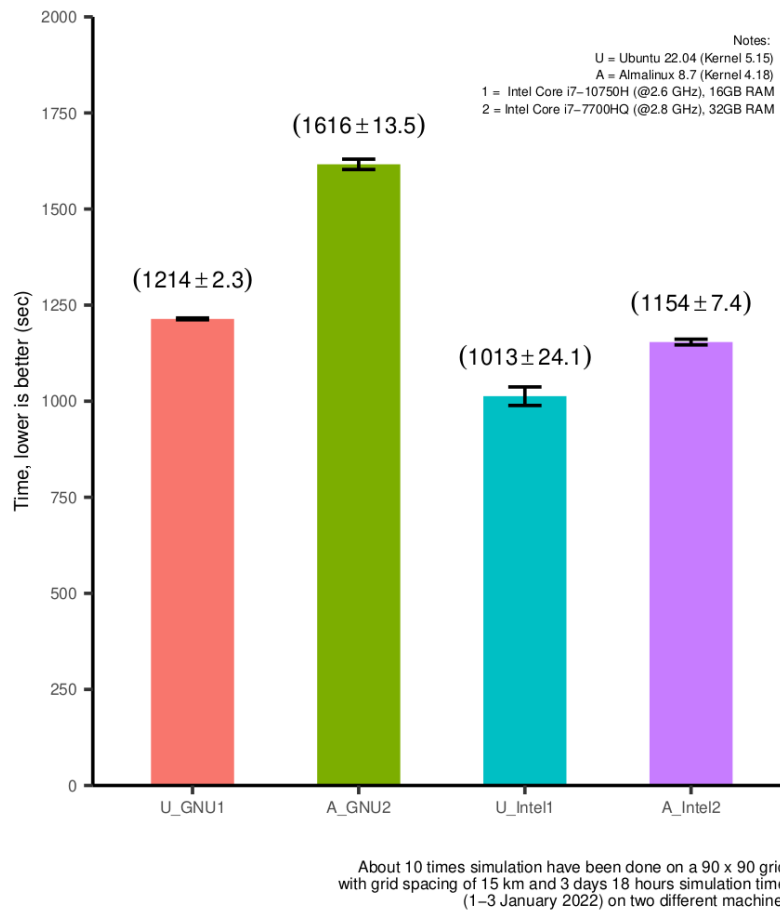
## 1 Dynamical Downscaling: *Model Weather Research Forecasting (WRF)*

yang tinggi, juga cakupan pemilihan wilayah simulasi. Proses perhitungan model dinamik maupun fisik akan lebih cepat jika menggunakan banyak prosesor atau yang bisa disebut dengan **komputasi paralel**. Kerangka kerja perangkat lunak WRF mendukung komputasi paralel yang efisien pada berbagai platform komputasi. Model atmosfer membawa kumpulan komputasi yang sama di setiap grid vertikal maupun horizontal Gambar 1.2. Komputer melakukan perhitungan fisik atmosfer untuk setiap grid secara paralel. Kita mungkin telah mengenal jenis prosesor yang tersedia saat ini, seperti Intel dan AMD. Jenis prosesor yang mereka buat dapat mempengaruhi kecepatan proses perhitungan. Untuk perhitungan paralel, Anda dapat menggunakan tipe prosesor desktop, mobile (prosesor di laptop), maupun server, tetapi juga perlu memperhatikan banyaknya *core* maupun frekuensi yang digunakan.



Gambar 1.2: Ilustrasi Model Iklim

Terkait dengan bahasa pemrograman yang digunakan di dalam WRF, yaitu bahasa Fortran dan C, kode skrip terlebih dahulu harus dikompilasi dengan program Compiler. Compiler ini bertujuan mengubah bahasa dari kode yang diketik menjadi bahasa mesin. Ada berbagai macam program Compiler yang tersedia secara gratis, salah satunya adalah [GNU Compiler Collection \(gcc\)](#). Selain GNU, perusahaan seperti Intel, AMD, maupun NVIDIA juga memiliki program Compiler yang dapat digunakan secara gratis serta mendukung komputasi paralel. Di dalam modul praktikum ini, Anda dapat memilih salah satu dari dua Compiler yang akan dijelaskan, yaitu GNU atau Intel. Perbedaan terhadap jenis Compiler ini dapat mempengaruhi waktu simulasi pada satu jenis prosesor yang sama Gambar 1.3.



Gambar 1.3: Perbedaan waktu simulasi WRF pada tipe Compiler, Distro, dan Prosesor yang digunakan

## 1.2 Dasar-dasar Pemrograman Bash

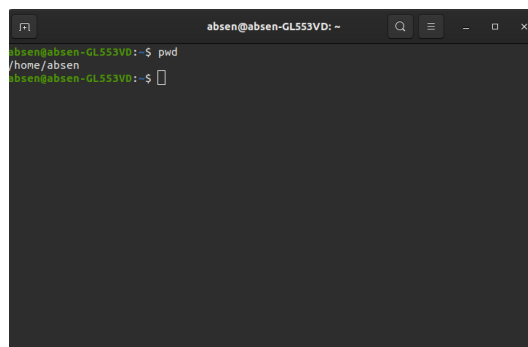
Sebagai salah satu tool scripting populer pada UNIX, Bash sangatlah berguna bagi para user Linux atau SysAdmin. Nama Bash merupakan akronim dari *Bourne Again Shell*. Sebagian besar model iklim, Bash digunakan untuk menjalankan instalasi dan simulasi. Pengguna model-model iklim disarankan mempelajari dasar-dasar pemrograman ini agar memahami berbagai perintah dari cara kerja instalasi dan simulasi model tersebut, biasanya terdapat file bernama **README**. File ini berisi mengenai cara instalasi, simulasi, informasi file, dan berbagai aplikasi tambahan untuk menjalankan model iklim tersebut.

Bash merupakan shell default dari sistem operasi Linux. Anda bisa menjalankan shell lain, seperti ksh (Korn Shell), zsh, dan csh (C shell). Untuk menjalankan Bash, Anda bisa menekan Ctrl+Alt+T atau carilah aplikasi Terminal. Khusus Windows 10, Anda dapat mencari aplikasi “Ubuntu” atau “wsl.exe”. Tampilan Bash seperti pada Gambar 1.4. Tulisan `absenabsen-GL553VD` menunjukkan nama user dan tanda `~` berarti lokasi folder saat ini.



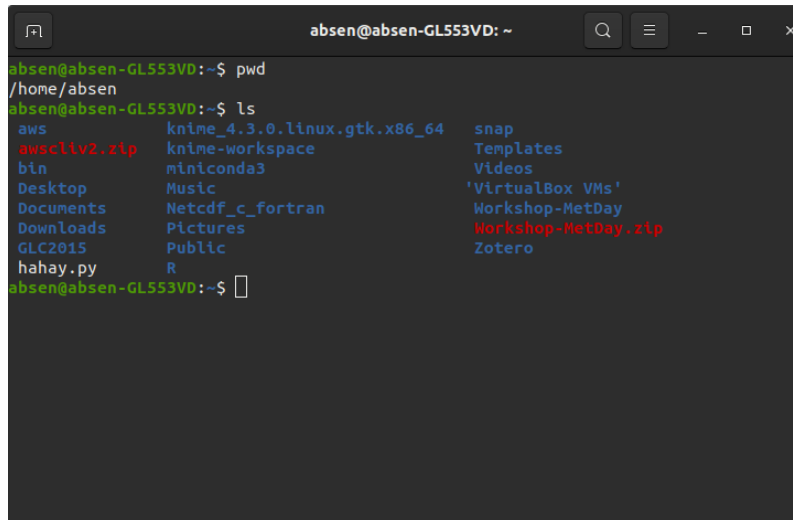
Gambar 1.4: Bash pada Ubuntu

Jika Anda mengetik perintah `pwd` pada Bash, hasilnya adalah seperti di bawah ini atau bisa dilihat pada Gambar 1.5.



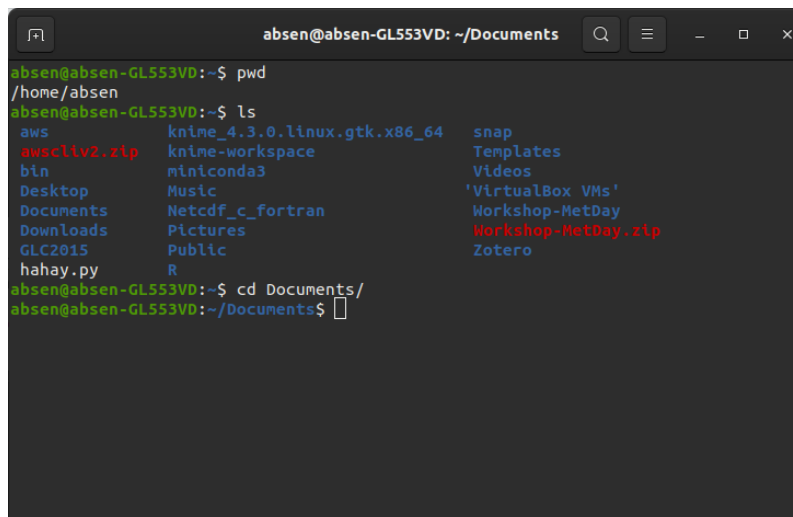
Gambar 1.5: Perintah `pwd` pada terminal Bash

Perintah dasar lainnya yang sering digunakan oleh pengguna Linux adalah `ls` atau `ll` (Gambar 1.6). Perintah ini berfungsi untuk melihat nama folder dan file yang berada di direktori pada lokasi saat ini. Untuk berpindah lokasi folder, Anda dapat mengetik perintah `cd` dan pilih folder yang ingin dibuka, contohnya `cd Documents/` atau `cd Documents` (Gambar 1.7). Anda dapat kembali ke folder sebelumnya dengan mengetik perintah `cd ..`



```
absen@absen-GL553VD: ~$ pwd
/home/absen
absen@absen-GL553VD:~$ ls
aws          knime_4.3.0.linux.gtk.x86_64  snap
awscliv2.zip knime-workspace               Templates
bin          miniconda3                   Videos
Desktop      Music                        'VirtualBox VMs'
Documents    Netcdf_c_fortran             Workshop-MetDay
Downloads    Pictures                     Workshop-MetDay.zip
GLC2015      Public                       Zotero
hahay.py     R
absen@absen-GL553VD:~$
```

Gambar 1.6: Perintah `ls` pada terminal Bash



```
absen@absen-GL553VD: ~/Documents$ pwd
/home/absen
absen@absen-GL553VD:~$ ls
aws          knime_4.3.0.linux.gtk.x86_64  snap
awscliv2.zip knime-workspace               Templates
bin          miniconda3                   Videos
Desktop      Music                        'VirtualBox VMs'
Documents    Netcdf_c_fortran             Workshop-MetDay
Downloads    Pictures                     Workshop-MetDay.zip
GLC2015      Public                       Zotero
hahay.py     R
absen@absen-GL553VD:~$ cd Documents/
absen@absen-GL553VD:~/Documents$
```

Gambar 1.7: Perintah `cd` pada terminal Bash

Anda dapat membuat folder baru dengan perintah `mkdir`, memindahkan folder atau file dengan `mv`, menghapus file dengan `rm`, menghapus folder dengan `rmdir`, membuat file baru dengan `touch`, dan menyalin folder atau file dengan `cp`. Contoh penggunaan beberapa perintah tersebut dapat dilihat sebagai berikut.

- Misalkan kita ingin membuat folder “Tes-folder”

## 1 Dynamical Downscaling: *Model Weather Research Forecasting (WRF)*

```
mkdir Tes-folder
```

Anda juga dapat membuat lebih dari 1 folder, misalnya “Tes-folder-1” dan “Tes-folder-2”

```
mkdir Tes-folder-1 Tes-folder-2
```

Untuk memastikan ketiga folder tersebut telah dibuat, ketik perintah `ls`

```
ls
```

- Misalkan kita ingin membuat file “tes-file” di dalam folder “Tes-folder”

```
cd Tes-folder  
touch tes-file  
ls
```

Kemudian, file “tes-file” kita pindahkan ke direktori sebelumnya dan ganti namanya dengan “tes-file-pindah”

```
mv tes-file ../tes-file-pindah
```

Kembali ke folder sebelumnya

```
cd ..
```

Ketik perintah `ls` untuk memastikan file “tes-file” telah dipindahkan ke direktori sebelumnya dengan berubah nama menjadi “tes-file-pindah”

```
ls
```

- File “tes-file-pindah” dihapus menggunakan perintah `rm`

```
rm tes-file-pindah  
ls
```

## 1.3 Prosedur Praktikum: Instalasi WRF

Kami menyarankan untuk menggunakan sistem operasi Linux dengan Distro Ubuntu. Jika Anda menggunakan Windows 10/11, Anda dapat memasang Windows Subsystem Linux (WSL) dan pasang Distro Ubuntu. Saat kami menguji WRF di Ubuntu, proses instalasi serta simulasi tidak mengalami masalah. Anda dipersilahkan menggunakan distro yang lain, seperti Fedora, CentOS, Almalinux, atau Manjaro, tetapi kami tidak dapat menjamin keberhasilan instalasi dan simulasi pada distro tersebut. Untuk pengguna MacOS, Anda dapat menggunakan Virtual Machine (mis. Virtual Box, VMWare, QEMU) atau Docker.

### 1.3.1 Menjalankan *Windows Subsystem Linux (WSL)* di Windows 10/11

Untuk menjalankan WRF, Anda memerlukan sistem operasi berbasis Linux dengan distribusi yang tersedia saat ini. Pada modul ini, distro Ubuntu yang dipilih untuk simulasi WRF. Bagi Anda pengguna Windows 10/11, WSL dapat digunakan. Selain WSL, Anda juga dapat menggunakan Cygwin. WSL dapat dijalankan pada Windows 10/11 dan Windows Server 2019 dengan versi minimum 1803. Saat ini, WSL versi 2 (WSL-2) telah tersedia yang dapat berjalan pada Windows 10 versi 1903 ke atas. Kami merekomendasikan Anda untuk memasang WSL-2 daripada WSL-1 karena prosesnya lebih cepat. Untuk mengaktifkan WSL, Anda dapat mengikuti beberapa langkah-langkah berikut ini.

1. Unduh WSL-2 pada [halaman ini](#) dan Install.
2. Buka menu Control Panel → Program and Features
3. Di sebelah kiri, klik Turn Windows Features On or Off
4. Scroll ke paling bawah. Kemudian, aktifkan tanda centang pada **Windows Subsystem for Linux** dan **Virtual Machine Platform**
5. Tunggu proses *update* sampai selesai, kemudian klik **Restart Now** (pastikan simpan data-data pekerjaan dan tutup semua aplikasi)
6. Setelah proses restart selesai, carilah aplikasi dan unduh Ubuntu pada Microsoft Store
7. Buka aplikasi Ubuntu yang telah terunduh dan tunggu konfigurasi selesai
8. Masukkan Username dan Password (**Catatan:** kami sarankan sama dengan user dan pass Windows 10, tapi boleh beda. Hasil ketikan password **tidak** muncul di terminal)
9. Ketikkan pada terminal

```
sudo apt update
```

10. Anda juga dapat membuka terminal Ubuntu pada terminal Windows Power Shell dengan mengetik `wsl` atau `wsl.exe`

### 1.3.2 Memasang NCAR Command Language (NCL)

Aplikasi ini dibutuhkan untuk memvisualisasikan lokasi kajian yang akan disimulasikan. **NCL** merupakan bahasa pemrograman interpreter, sama seperti Python, yang memiliki banyak kegunaan dalam proses analisis dan visualisasi data-data cuaca dan iklim. Pengembang WRF telah menyediakan skrip NCL untuk memudahkan dalam ketepatan pemilihan lokasi sesuai dengan keinginan pengguna. Langkah-langkah pemasangan NCL sebagai berikut.

1. Untuk memasang NCL, Anda harus mengunduh aplikasi Miniconda untuk Linux pada website [https://repo.anaconda.com/miniconda/Miniconda3-py39\\_22.11.1-1-Linux-x86\\_64.sh](https://repo.anaconda.com/miniconda/Miniconda3-py39_22.11.1-1-Linux-x86_64.sh) dengan perintah pada terminal Bash sebagai berikut.

```
wget https://repo.anaconda.com/miniconda/Miniconda3-py39_22.11.1-1-Linux-x86_64.sh
```

2. Kemudian, lakukan pemasangan Miniconda dengan perintah di bawah ini.

```
bash Miniconda3-py39_22.11.1-1-Linux-x86_64.sh
```

3. Selanjutnya tekan ENTER. Terminal akan menampilkan *End-User License Agreement (EULA)*, tekan ENTER atau SPACE sampai muncul perintah seperti di bawah ini.

```
Do you accept the license terms? [yes|no]
[no] >>>
```

4. Kemudian, ketikkan **yes** untuk melanjutkan instalasi dan tekan ENTER.
5. Tutup terminal Anda dan buka kembali. Perhatikan pada tulisan (**base**) di paling kiri nama user. Jika tulisan itu sudah muncul, berarti proses pemasangan Miniconda telah berhasil.
6. Selanjutnya, lakukan pemasangan package **mamba** dengan perintah.

```
conda install mamba -n base -c conda-forge
```

#### **i** Note

Package **mamba** memungkinkan dapat mempercepat pengunduhan dan pemasangan package-package di dalam Miniconda.

7. Terakhir, lakukan proses pembuatan *environment* dengan nama **ncl** dan pemasangan NCL dengan perintah.

```
conda create -n ncl
conda activate ncl
```



```
mamba install -c conda-forge xarray netcdf4 scipy pyngl pynio matplotlib cartopy  
↪ wrf-python ncl
```

8. Saat Anda ingin memulai menggunakan program NCL, aktifkan terlebih dahulu *environment ncl* dengan perintah di bawah ini. Kemudian, tulisan (base) menjadi (ncl) yang menandakan bahwa *environment* telah berhasil diaktifkan.

```
conda activate ncl
```

#### **i** Note

*Environment ncl* akan terus dipakai, mulai dari pemasangan WRF hingga analisis dan visualisasi luaran WRF. NCAR telah mengembangkan **wrf-python** secara khusus untuk analisis dan visualisasi luaran WRF.

### 1.3.3 Sebelum Mulai

Anda perlu mencoba dasar pemrograman Bash sebelum memulai mengunduh data, memasang beberapa package, sampai menjalankan simulasi WRF. Silahkan Anda ikuti langkah-langkah di bawah ini agar memudahkan dalam mencoba praktikum ini.

1. Buka terminal Bash

- Untuk Windows 10/11, buka aplikasi Windows Power Shell. Kemudian, ketikkan perintah seperti di bawah ini. Setelahnya, akan muncul terminal Bash.

```
wsl
```

- Untuk Ubuntu atau disto Linux lainnya, carilah program **Terminal** atau bisa menggunakan shortcut **Ctrl+Alt+T**
2. Saat Anda membuka terminal, pastikan lokasi folder/direktori saat ini adalah `/home/<user_name>`, di mana **user\_name** adalah nama user pada laptop/komputer Anda masing-masing. Perhatikan kembali Gambar 1.5 bahwa untuk melihat lokasi direktori Anda sekarang bisa mengetik perintah `pwd` pada terminal atau dengan melihat simbol `~` yang terletak di sebelah kiri `$`.
  3. Buat folder kerja dengan nama **WRF-Model** untuk menampung data-data, package, serta source code WRF. Perintah pada terminal Bash adalah sebagai berikut.

```
mkdir WRF-Model
```

4. Kemudian, bukalah folder **WRF-Model** dengan perintah

## 1 Dynamical Downscaling: *Model Weather Research Forecasting (WRF)*

```
cd WRF-Model
```

Sekarang, Anda berada di folder **WRF-Model** (Perhatikan ~ berubah menjadi ~/WRF-Model)

5. Di dalam **WRF-Model**, buatlah folder **data** guna untuk meletakkan data masukan WRF (data statik dan data cuaca).

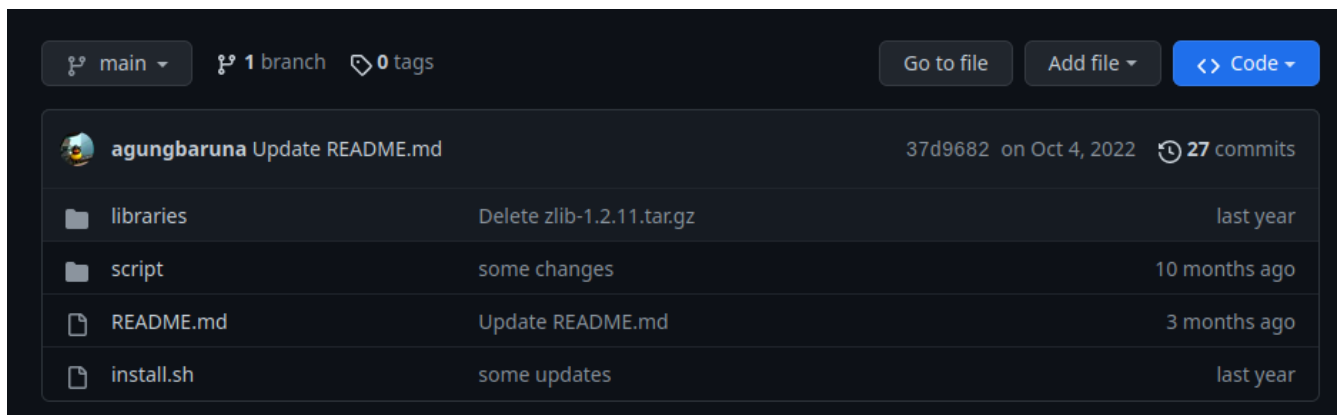
Langkah-langkah di atas masih berlanjut pada subbab selanjutnya.

### 1.3.4 Data dan Package Pendukung

#### Perangkat Lunak

Untuk memasang WRF, Anda perlu menyiapkan perangkat lunak pendukung serta data contoh untuk mensimulasikan WRF. Perangkat lunak tersebut dapat diunduh melalui halaman [Github ini](#). Untuk mengunduh source code, pilih <> **Code** → **Download ZIP** (Gambar 1.8). Anda dapat mengunduh secara manual atau menggunakan perintah `git clone` pada terminal.

```
sudo apt -y install git #Install program git jika belum ada
git clone https://www.github.com/agungbaruna/pyWRF-install
```



Gambar 1.8: Unduh source code

Setelah software pendukung telah diunduh, unduh source code WPS (WRF Pre-Processing) dan WRF pada halaman [Github NCAR](#). Source code WRF berisi algoritma perhitungan fisik dan dinamik atmosfer, sedangkan source code WPS berisi algoritma untuk persiapan data masukan atmosfer dari GCM dan pemilihan lokasi.

- WRF Pre-Processing (WPS): <https://www.github.com/wrf-model/WPS>
- Model WRF: [www.github.com/wrf-model/WRF](https://www.github.com/wrf-model/WRF)

## Data

Data-data masukan untuk WRF telah tersedia dan unduh pada halaman web <https://s.id/wrf-data>. Kami menyarankan untuk menggunakan Wi-Fi karena ukuran data cukup besar. Data yang telah diunduh Anda pindahkan ke folder `data` yang berada di dalam `WRF-Model`.

### 1. Global Forecast System (GFS)

Untuk data masukan yang digunakan berasal dari NOAA yang bernama Global Forecasts System (GFS). GFS memiliki resolusi spasial  $0.25^\circ$  ( $\sim 25$  km),  $0.50^\circ$  ( $\sim 50$  km), dan  $1.00^\circ$  ( $\sim 100$  km) dengan temporal per 6 jam. GFS memiliki produk data prakiraan maupun analisis/histori cuaca secara global. Anda dapat mengunduh data ini pada salah satu halaman web berikut.

- AWS S3 Bucket: <https://noaa-gfs-bdp-pds.s3.amazonaws.com>
- Research Data Archive (RDA) NCAR: <https://rda.ucar.edu/datasets/ds084.1>
- NCEP Central Operations: <https://nomads.ncep.noaa.gov/>

Data GFS yang digunakan dalam praktikum ini memiliki waktu 1-3 Januari 2022 pada pukul 00:00, 06:00, 12:00, dan 18:00 UTC dengan resolusi spasial  $1.00^\circ$  yang telah diunduh pada halaman web AWS S3 Bucket. Produk GFS yang digunakan adalah analisis/histori.

### 2. ERA5

Selain GFS, Anda dapat menggunakan data masukan dari institusi lain, seperti ECMWF pada produk ERA5. ERA5 merupakan data reanalisis sehingga hanya memiliki produk histori. Anda dapat mengunduhnya melalui Climate Data Store (CDS) pada halaman <https://cds.climate.copernicus.eu>. ERA5 memiliki resolusi spasial sebesar  $0.25^\circ$  dengan temporal per 1 jam. Data ERA5 memiliki dua tipe, yaitu data permukaan tanah (*Single Levels*) dan atmosfer untuk setiap ketinggian (*Pressure Levels*). Anda diharuskan mengunduh dua tipe data ini dengan variabel yang dipilih adalah sebagai berikut.

- ERA5 hourly data on Pressure Levels:
  - geopotential
  - relative humidity
  - specific humidity
  - temperature
  - u-component wind
  - v-component wind
- ERA5 hourly data on Single Levels:
  - 10m u-component of wind
  - 10m v-component of wind
  - 2m dewpoint temperature
  - 2m temperature
  - land sea mask
  - mean sea level pressure

## 1 Dynamical Downscaling: *Model Weather Research Forecasting (WRF)*

- sea ice cover
- sea surface temperature
- skin temperature
- snow depth
- soil temperature level 1
- soil temperature level 2
- soil temperature level 3
- soil temperature level 4
- surface\_pressure
- volumetric soil water layer 1
- volumetric soil water layer 2
- volumetric soil water layer 3
- volumetric soil water layer 4

Anda tidak diharuskan mengunduh semua jam karena CDS membatasi banyaknya permintaan data dari pengguna sehingga Anda tidak dapat mengunduh semua data sekaligus. Data ERA5 yang akan digunakan pada praktikum ini memiliki waktu yang sama dengan GFS.

### 3. Data Statik WPS

Data statik ini wajib diunduh untuk menjalankan WRF. Data ini berisi seperti tipe permukaan lahan, nilai Leaf Area Index (LAI), tipe tanah, dan elevasi permukaan. Anda dapat mengunduhnya di [https://www2.mmm.ucar.edu/wrf/src/wps\\_files/geog\\_high\\_res\\_mandatory.tar.gz](https://www2.mmm.ucar.edu/wrf/src/wps_files/geog_high_res_mandatory.tar.gz). File dari data tersebut berukuran 2 GB dan setelah diekstrak dapat mencapai 30 GB. Data ini sudah tersedia di dalam link <https://s.id/wrf-data>. Untuk mengekstrak file ini, gunakan perintah berikut

```
cd data
gunzip geog_high_res_mandatory.tar.gz
tar -xf geog_high_res_mandatory.tar
cd .. # Kembali ke folder WRF-Model
```

Setelah itu, folder WPS\_GEOG akan muncul di dalam folder data/

### 1.3.5 Instalasi Software Compiler

Instalasi ini meliputi cara melakukan kompilasi dengan dua Compiler berbeda, yaitu GNU dan Intel. Anda dipersilahkan memilih salah satu langkah. Proses kompilasi untuk semua package yang telah terunduh tidak dapat dilakukan pada Compiler yang berbeda.

#### 1. GNU

Sebelum Anda memasuki tahapan instalasi package yang terdapat dalam folder `pyWRF-install`, pasang terlebih dahulu package dependencies dengan perintah di bawah ini.

```
sudo apt -y update && sudo apt -y upgrade
sudo apt -y install gfortran gcc make m4 csh
```

## 2. Intel

Berbeda dengan GNU, tipe Compiler Intel memang agak rumit untuk memasangnya. Namun, Compiler ini sebenarnya lebih efisien dalam hal kecepatan saat mensimulasikan WRF. Kami sudah membandingkan waktu simulasi dengan Compiler tipe ini lebih cepat dibandingkan dengan GNU Gambar 1.3.

Anda dapat memperoleh software ini di halaman resmi [Intel](https://www.intel.com). Kami sudah menyediakan software tersebut di dalam halaman web [s.id/wrf-intel-compiler](https://www.intel.com/content/www/us/en/developer/tools/oneapi/wrf-intel-compiler.html). Terdapat tiga file instalasi Compiler, yaitu Fortran, C, dan MPI (untuk komputasi paralel). Jumlah ukuran tiga file tersebut cukup besar, sekitar 2,5 GB. Letakkan ketiga file ini di dalam folder WRF-Model. Perhatikan langkah-langkah berikut ini untuk memasang ketiga file tersebut.

```
# 0. Package pendukung
sudo apt -y install make m4 csh
# 1. C Compiler
bash l_dpcpp-cpp-compiler_p_2023.0.0.25393_offline.sh -a -s --eula accept
# 2. Fortran Compiler
bash l_fortran-compiler_p_2023.0.0.25394_offline.sh -a -s --eula accept
# 3. MPI
bash l_mpi_oneapi_p_2021.8.0.25329_offline.sh -a -s --eula accept
```

Lokasi folder hasil instalasi terdapat di `$HOME/intel`. Perintah untuk memanggil program Compiler Intel belum menjadi **ENVIRONMENT VARIABLE**. Untungnya, Intel menyediakan skrip untuk memanggil semua program Compiler secara default, yaitu terdapat di dalam `$HOME/intel/oneapi/setvars.sh`. Ketika memanggil skrip tersebut, **ENVIRONMENT VARIABLE** milik Miniconda secara otomatis dinonaktifkan (tulisan `<base>` hilang). Perintah untuk memanggil skrip `setvars.sh` adalah sebagai berikut.

```
source $HOME/intel/oneapi/setvars.sh
```

### 1.3.6 Instalasi Package

Instalasi beberapa package meliputi zlib, libpng, jasper, hdf5, netcdf-c, dan netcdf-fortran. Perlu diperhatikan dan lihat pada terminal bahwa Anda berada di folder `~/WRF-Model`. Masukkan beberapa perintah **ENVIRONMENT VARIABLE** di bawah ini dengan perintah `export`.

```
export ODIR=$HOME/WRF-Model
export PATH=$ODIR/bin:$PATH
```

```
export LD_LIBRARY_PATH=$ODIR/lib:$LD_LIBRARY_PATH
export LDFLAGS=-L$ODIR/lib
export CPPFLAGS=-I$ODIR/include
export NETCDF=$ODIR
export HDF5=$ODIR
export JASPERLIB=$ODIR/lib
export JASPERINC=$ODIR/include
```

Langkah ini wajib dilakukan pada saat melakukan instalasi package dan juga WRF. Pendefinisian **ENVIRONMENT VARIABLE** ini berlaku untuk GNU maupun Intel.

## 1. GNU

Berikut ini langkah-langkah memasang package pada Compiler **GNU**. Buka terlebih dahulu folder `pyWRF-install/libraries` dengan mengetik perintah `cd pyWRF-install/libraries`.

### 1. zlib

```
# 0. Buka folder libraries
cd $ODIR/pyWRF-install/libraries
# 1. Extract
tar -xf zlib-1.2.13.tar.gz
# 2. Buka folder hasil ekstrak
cd zlib-1.2.13
# 3. Konfigurasi
CC=gcc ./configure --prefix=$ODIR
# 4. Instalasi
make check install
```

### 2. libpng

```
# 0. Buka folder libraries
cd $ODIR/pyWRF-install/libraries
# 1. Extract
tar -xf libpng-1.6.37.tar.gz
# 2. Buka folder hasil ekstrak
cd libpng-1.6.37
# 3. Konfigurasi
CC=gcc ./configure --prefix=$ODIR
# 4. Instalasi
make check install
```

### 3. jasper

```
# 0. Buka folder libraries
cd $ODIR/pyWRF-install/libraries
# 1. Extract
tar -xf jasper-1.900.1.tar.gz
# 2. Buka folder hasil ekstrak
cd jasper-1.900.1
# 3. Konfigurasi
CC=gcc ./configure --prefix=$ODIR
# 4. Instalasi
make check install
```

#### 4. openMPI

Jika Anda tidak berencana menjalankan WRF dengan 1 prosesor, Anda dapat melewati langkah ini. Akan tetapi, kami menyarankan untuk memasang program ini untuk mempersingkat waktu dalam mensimulasikan WRF.

```
# 0. Buka folder libraries
cd $ODIR/pyWRF-install/libraries
# 1. Extract
tar -xf openmpi-4.1.4.tar.gz
# 2. Buka folder hasil ekstrak
cd openmpi-4.1.4
# 3. Konfigurasi
CC=gcc FC=gfortran ./configure --prefix=$ODIR
# 4. Instalasi
make check install
```

#### 5. hdf5

```
# 0. Buka folder libraries
cd $ODIR/pyWRF-install/libraries
# 1. Extract
tar -xf hdf5-1.12.0.tar.gz
# 2. Buka folder hasil ekstrak
cd hdf5-1.12.0
# 3. Konfigurasi
CC=gcc FC=gfortran ./configure --prefix=$ODIR --with-default-api-version=v18
↪ --enable-fortran
# 4. Instalasi
make
make install
```

#### 6. netcdf-c

## 1 Dynamical Downscaling: *Model Weather Research Forecasting (WRF)*

```
# 0. Buka folder libraries
cd $ODIR/pyWRF-install/libraries
# 1. Extract
tar -xf netcdf-c-4.7.4.tar.gz
# 2. Buka folder hasil ekstrak
cd netcdf-c-4.7.4
# 3. Konfigurasi
CC=gcc ./configure --prefix=$ODIR --disable-dap
# 4. Instalasi
make
make install
```

### 7. netcdf-fortran

```
# 0. Buka folder libraries
cd $ODIR/pyWRF-install/libraries
# 1. Extract
tar -xf netcdf-fortran-4.5.3.tar.gz
# 2. Buka folder hasil ekstrak
cd netcdf-fortran-4.5.3
# 3. Konfigurasi
CC=gcc FC=gfortran ./configure --prefix=$ODIR
# 4. Instalasi
make
make install
```

## 2. Intel

Berikut ini langkah-langkah memasang package pada Compiler **Intel**. Buka terlebih dahulu folder `pyWRF-install/libraries` dengan mengetik perintah `cd pyWRF-install/libraries`. Untuk proses instalasi menggunakan Intel, sama saja dengan GNU. Hanya saja, perbedaannya adalah definisi dari variabel `CC` maupun `FC`. Untuk Intel, variabel `CC=icc` dan `FC=ifort`. Jangan lupa untuk mengaktifkan **ENVIRONMENT VARIABLE** dari Intel dengan mengetik perintah.

```
source ~/intel/oneapi/setvars.sh
```

### 1. zlib

```
# 0. Buka folder libraries
cd $ODIR/pyWRF-install/libraries
# 1. Extract
tar -xf zlib-1.2.13.tar.gz
```



```
# 2. Buka folder hasil ekstrak
cd zlib-1.2.13
# 3. Konfigurasi
CC=icc CFLAGS='-diag-disable=10441' ./configure --prefix=$ODIR
# 4. Instalasi
make check install
```

## 2. libpng

```
# 0. Buka folder libraries
cd $ODIR/pyWRF-install/libraries
# 1. Extract
tar -xf libpng-1.6.37.tar.gz
# 2. Buka folder hasil ekstrak
cd libpng-1.6.37
# 3. Konfigurasi
CC=icc CFLAGS='-diag-disable=10441' ./configure --prefix=$ODIR
# 4. Instalasi
make check install
```

## 3. jasper

```
# 0. Buka folder libraries
cd $ODIR/pyWRF-install/libraries
# 1. Extract
tar -xf jasper-1.900.1.tar.gz
# 2. Buka folder hasil ekstrak
cd jasper-1.900.1
# 3. Konfigurasi
CC=icc CFLAGS='-diag-disable=10441' ./configure --prefix=$ODIR
# 4. Instalasi
make check install
```

## 4. hdf5

```
# 0. Buka folder libraries
cd $ODIR/pyWRF-install/libraries
# 1. Extract
tar -xf hdf5-1.12.0.tar.gz
# 2. Buka folder hasil ekstrak
cd hdf5-1.12.0
# 3. Konfigurasi
```

## 1 Dynamical Downscaling: Model Weather Research Forecasting (WRF)

```
CC=icc FC=ifort CFLAGS='-diag-disable=10441' ./configure --prefix=$ODIR  
↪ --with-default-api-version=v18 --enable-fortran  
# 4. Instalasi  
make  
make install
```

### 5. netcdf-c

```
# 0. Buka folder libraries  
cd $ODIR/pyWRF-install/libraries  
# 1. Extract  
tar -xf netcdf-c-4.7.4.tar.gz  
# 2. Buka folder hasil ekstrak  
cd netcdf-c-4.7.4  
# 3. Konfigurasi  
CC=icc CFLAGS='-diag-disable=10441' ./configure --prefix=$ODIR --disable-dap  
# 4. Instalasi  
make  
make install
```

### 6. netcdf-fortran

```
# 0. Buka folder libraries  
cd $ODIR/pyWRF-install/libraries  
# 1. Extract  
tar -xf netcdf-fortran-4.5.3.tar.gz  
# 2. Buka folder hasil ekstrak  
cd netcdf-fortran-4.5.3  
# 3. Konfigurasi  
CC=icc FC=ifort CFLAGS='-diag-disable=10441' ./configure --prefix=$ODIR  
# 4. Instalasi  
make  
make install
```

### 1.3.7 Instalasi WRF

Proses instalasi WRF membutuhkan waktu agak lama, yaitu sekitar 30-60 menit. Tentunya, ini bergantung pada spesifikasi prosesor yang Anda gunakan, serta tipe Compiler. Ikuti langkah-langkah berikut.

1. Buka direktori WRF yang berada di dalam \$ODIR/WRF dengan perintah.

```
cd $ODIR/WRF
```

2. Jalankan file `configure` dengan perintah.

```
./configure
```

Anda akan disajikan beberapa teks di dalamnya. Anda disuruh untuk memilih opsi Compiler. Tipe Compiler tersebut selain GNU dan Intel, ada pula IBM, PGI, Fujitsu, Pathscale, dan CRAY. Jika Anda menggunakan Compiler tipe GNU, ketikkan angka **35**. Untuk Intel, ketik angka **16**. Setelah itu, tekan Enter.

```

$ ./configure
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /home/absen/WRF-Model
Will use HDF5 in dir: /home/absen/WRF-Model
PHDF5 not set in environment. Will configure WRF for use without.
Will use 'time' to report timing information

If you REALLY want Grib2 output from WRF, modify the arch/Config.pl script.
Right now you are not getting the Jasper lib, from the environment, compiled into WRF.

-----
Please select from among the following Linux x86_64 options:

1. (serial) 2. (smpar) 3. (dmpar) 4. (dmsm) PGI (pgf90/gcc)
5. (serial) 6. (smpar) 7. (dmpar) 8. (dmsm) PGI (pgf90/gcc): SGI MPT
9. (serial) 10. (smpar) 11. (dmpar) 12. (dmsm) PGI (pgf90/gcc): PGI accelerator
13. (serial) 14. (smpar) 15. (dmpar) 16. (dmsm) INTEL (ifort/icc)
17. (dmsm) INTEL (ifort/icc): Xeon Phi (MIC architecture)
18. (serial) 19. (smpar) 20. (dmpar) 21. (dmsm) INTEL (ifort/icc): Xeon (SMB with AVX mods)
22. (serial) 23. (smpar) 24. (dmpar) 25. (dmsm) INTEL (ifort/icc): SGI MPT
26. (serial) 27. (smpar) 28. (dmpar) 29. (dmsm) INTEL (ifort/icc): IBM POE
30. (serial) 31. (dmpar) PATHSCALE (path90/pathcc)
32. (serial) 33. (smpar) 34. (dmpar) 35. (dmsm) GNU (gfortran/gcc)
36. (serial) 37. (smpar) 38. (dmpar) 39. (dmsm) IBM (xlf90_r/cc_r)
40. (serial) 41. (smpar) 42. (dmpar) 43. (dmsm) PGI (ftn/gcc): Cray XC CLE
44. (serial) 45. (smpar) 46. (dmpar) 47. (dmsm) CRAY CCE (ftn SNOBOL/cc): Cray XE and XC
48. (serial) 49. (smpar) 50. (dmpar) 51. (dmsm) INTEL (ftn/icc): Cray XC
52. (serial) 53. (smpar) 54. (dmpar) 55. (dmsm) PGI (pgf90/gcc)
56. (serial) 57. (smpar) 58. (dmpar) 59. (dmsm) PGI (pgf90/gcc): -f90pgf90
60. (serial) 61. (smpar) 62. (dmpar) 63. (dmsm) PGI (pgf90/gcc): -f90pgf90
64. (serial) 65. (smpar) 66. (dmpar) 67. (dmsm) INTEL (ifort/icc): HSW/BDW
68. (serial) 69. (smpar) 70. (dmpar) 71. (dmsm) INTEL (ifort/icc): KNL MIC
72. (serial) 73. (smpar) 74. (dmpar) 75. (dmsm) FUJITSU (frtpr/fccpx): FX10/FX100 SPARC64 Ix/fx/Xl/fx

Enter selection [1-75] : 35

```

Gambar 1.9: Konfigurasi WRF dengan Compiler GNU

```

$ ./configure
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /home/absen/WRF-Model
Will use HDF5 in dir: /home/absen/WRF-Model
PHDF5 not set in environment. Will configure WRF for use without.
Will use 'time' to report timing information

If you REALLY want Grib2 output from WRF, modify the arch/Config.pl script.
Right now you are not getting the Jasper lib, from the environment, compiled into WRF.

-----
Please select from among the following Linux x86_64 options:

1. (serial) 2. (smpar) 3. (dmpar) 4. (dmsm) PGI (pgf90/gcc)
5. (serial) 6. (smpar) 7. (dmpar) 8. (dmsm) PGI (pgf90/gcc): SGI MPT
9. (serial) 10. (smpar) 11. (dmpar) 12. (dmsm) PGI (pgf90/gcc): PGI accelerator
13. (serial) 14. (smpar) 15. (dmpar) 16. (dmsm) INTEL (ifort/icc)
17. (dmsm) INTEL (ifort/icc): Xeon Phi (MIC architecture)
18. (serial) 19. (smpar) 20. (dmpar) 21. (dmsm) INTEL (ifort/icc): Xeon (SMB with AVX mods)
22. (serial) 23. (smpar) 24. (dmpar) 25. (dmsm) INTEL (ifort/icc): SGI MPT
26. (serial) 27. (smpar) 28. (dmpar) 29. (dmsm) INTEL (ifort/icc): IBM POE
30. (serial) 31. (dmpar) PATHSCALE (path90/pathcc)
32. (serial) 33. (smpar) 34. (dmpar) 35. (dmsm) GNU (gfortran/gcc)
36. (serial) 37. (smpar) 38. (dmpar) 39. (dmsm) IBM (xlf90_r/cc_r)
40. (serial) 41. (smpar) 42. (dmpar) 43. (dmsm) PGI (ftn/gcc): Cray XC CLE
44. (serial) 45. (smpar) 46. (dmpar) 47. (dmsm) CRAY CCE (ftn SNOBOL/cc): Cray XE and XC
48. (serial) 49. (smpar) 50. (dmpar) 51. (dmsm) INTEL (ftn/icc): Cray XC
52. (serial) 53. (smpar) 54. (dmpar) 55. (dmsm) PGI (pgf90/gcc)
56. (serial) 57. (smpar) 58. (dmpar) 59. (dmsm) PGI (pgf90/gcc): -f90pgf90
60. (serial) 61. (smpar) 62. (dmpar) 63. (dmsm) PGI (pgf90/gcc): -f90pgf90
64. (serial) 65. (smpar) 66. (dmpar) 67. (dmsm) INTEL (ifort/icc): HSW/BDW
68. (serial) 69. (smpar) 70. (dmpar) 71. (dmsm) INTEL (ifort/icc): KNL MIC
72. (serial) 73. (smpar) 74. (dmpar) 75. (dmsm) FUJITSU (frtpr/fccpx): FX10/FX100 SPARC64 Ix/fx/Xl/fx

Enter selection [1-75] : 16

```

Gambar 1.10: Konfigurasi WRF dengan Compiler GNU

3. Untuk melakukan instalasi, ketik perintah ini.

```
./compile em_real -j jumlah_prosesor
```

dimana `jumlah_prosesor` adalah jumlah dari prosesor pada laptop/komputer Anda yang akan digunakan untuk proses instalasi dan kompilasi kode-kode WRF. Proses kompilasi akan memakan

## 1 Dynamical Downscaling: *Model Weather Research Forecasting (WRF)*

waktu yang sangat lama apabila Anda hanya menggunakan 1 prosesor. Pastikan berbagai program pada komputer/laptop Anda yang saat ini sedang dibuka, seperti Google Chrome atau Spotify harap ditutup terlebih dahulu karena ini membantu proses instalasi lebih stabil.

4. **Untuk Compiler Intel**, setelah langkah ke-2 dijalankan, buka file `configure.wrf` dengan perintah.

```
nano configure.wrf
```

Kemudian, scroll ke bawah dengan menekan tombol ↓ pada keyboard dan ubahlah isinya sesuai aturan pada tabel di bawah ini.

Tabel 1.1: Pengubahan variabel DM\_FC dan DM\_CC

Sebelum	Sesudah
DM_FC = mpif90 -f90=\$(SFC)	DM_FC = mpiifort -f90=\$(SFC)
DM_CC = mpicc -cc=\$(SCC)	DM_CC = mpiicc -cc=\$(SCC)

Setelah selesai diubah, keluar dari editor nano dengan menekan tombol **Ctrl + X**

Proses instalasi WRF berhasil dilakukan apabila terdapat file yang berekstensi `.exe`: `ndown.exe`, `tc.exe`, `real.exe`, dan `wrf.exe` di dalam folder `main`. Anda bisa melihatnya dengan perintah

```
ls main/*.exe
```

```
main/ndown.exe  main/real.exe  main/tc.exe  main/wrf.exe
```

### 1.3.8 Instalasi WRF Pre-Processing (WPS)

Program WPS digunakan untuk menyesuaikan data masukan dari berbagai sumber (ECMWF, GFS, NAM, ...) sebelum ke simulasi WRF. Terdapat 3 program utama: `geogrid.exe`, `ungrib.exe`, dan `metgrid.exe`. Berikut ini kegunaan dari ketiga program utama WPS.

- **geogrid.exe**: memilih lokasi yang akan dilakukan simulasi. Luaran program ini berupa file `geo_em*` yang berisi nilai-nilai dari variabel di dalam file hasil ekstrak `geog_high_res_mandatory.tar`
- **ungrib.exe**: mengubah file berformat `grib` menjadi `nc`, serta memungkinkan dapat melakukan interpolasi (waktu dan lokasi)
- **metgrid.exe**: menggabungkan

Proses instalasi WPS tidak membutuhkan waktu yang lama, sekitar 2-5 menit. Untuk melakukan instalasi WPS, ikuti langkah-langkah berikut ini.

## 1.4 Prosedur Praktikum: Menjalankan Simulasi WRF-ARW

1. Saat ini, Anda berada di folder WRF. Buka folder WPS dengan perintah

```
cd $ODIR/WPS
```

2. Jalankan file `configure` dengan perintah

```
./configure
```

Ketik angka **3** untuk GNU atau angka **19** untuk Intel.

3. Setelah selesai, di folder WPS akan muncul file `configure.wps`. Beberapa baris dari isi file tersebut ada yang perlu ditambahkan dan diganti dengan ketentuan ini.

- Untuk Intel, tambahkan *flags -liomp5* setelah `-lnetcdf` pada bagian variabel `WRF_LIB = ....`. Kemudian, ubah pula `DM_FC` dan `DM_CC` seperti pada Tabel 1.1.
- Untuk GNU, tambahkan *flags -fopenmp* setelah `-lnetcdf` pada bagian variabel `WRF_LIB = ....`

4. Setelah diganti, lakukan kompilasi dengan mengetik perintah

```
./compile
```

Proses instalasi berhasil apabila terdapat 3 file `.exe`: `geogrid.exe`, `ungrib.exe`, dan `metgrid.exe` di folder WPS yang berupa shortcut. Anda dapat menggunakan perintah

```
ls *.exe
```

```
geogrid.exe metgrid.exe ungrib.exe
```

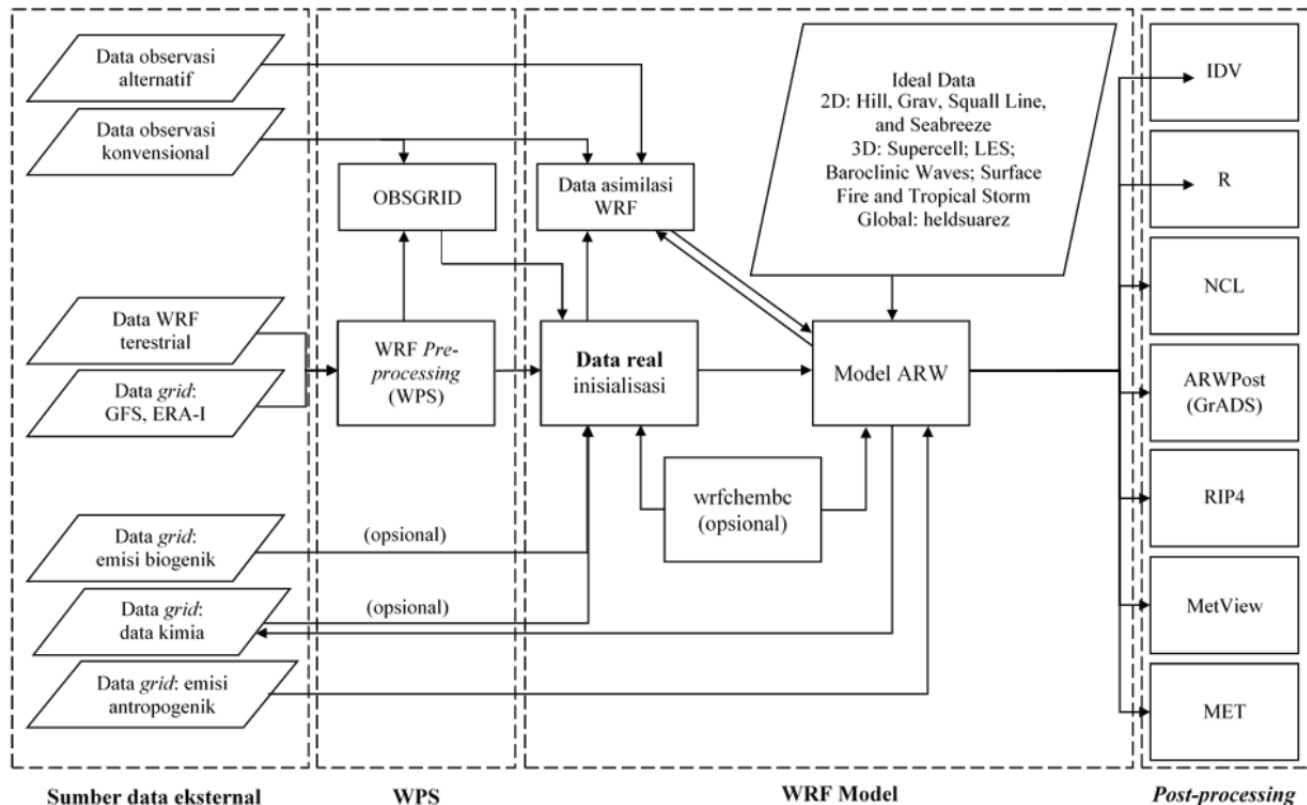
## 1.4 Prosedur Praktikum: Menjalankan Simulasi WRF-ARW

Anda dapat melanjutkan ke tahapan ini apabila seluruh program telah berhasil terpasang. Bagi Anda yang belum berhasil, sabar :D dan ulangi kembali langkah-langkah di atas. Secara umum, diagram pada Gambar 1.11 menunjukkan proses menjalankan WRF dari tahapan memasukkan data hingga plot dan analisis akhir. Seperti yang telah dijelaskan pada subbab sebelumnya bahwa langkah awal dalam menjalankan WRF adalah menjalankan WPS terlebih dahulu. Anda perlu menyiapkan data masukan atmosfer maupun permukaan (data statik). Program `geogrid.exe` dijalankan pertama, kemudian diikuti dengan `ungrib.exe` dan terakhir `metgrid.exe`. Selanjutnya, Anda dapat melangkah ke program WRF, yaitu `real.exe` dan `wrf.exe`. Program `real.exe` digunakan sebagai pendefinisian kondisi awal dan kondisi batas berdasarkan informasi dari `namelist.input` yang berada dalam folder `test/em_real/`. Kemudian, Anda dapat menggunakan perangkat lunak apapun (mis. NCAR Command Language (NCL), GrADS, R, Python, Matlab, ArcMAP, QGIS, ...) untuk menganalisis serta visualisasi luaran WRF. NCL

## 1 Dynamical Downscaling: Model Weather Research Forecasting (WRF)

yang telah Anda pasang sebelumnya dapat digunakan untuk analisis, tetapi kami hanya memberikan panduan analisis luaran WRF menggunakan Python.

Untuk data masukan, Anda juga dapat menggunakan data observasi, tetapi harus berupa grid. Namun, ini merupakan program yang berbeda dari WRF-ARW, yaitu WRFDA (WRF Data Assimilation). Selain itu, terdapat pula data masukan dari emisi kimia, seperti emisi biogenik dan antropogenik. Ini juga merupakan program yang berbeda dari WRF-ARW, yaitu WRF-Chem. Penjelasan mengenai WRFDA dan WRF-Chem tidak disampaikan di dalam praktikum ini.



Gambar 1.11: Diagram WRF

### 1.4.1 Program WPS

Untuk lebih mudah dalam memahami alur proses simulasi WRF pada modul ini, lokasi/domain yang dipilih untuk adalah Kota Surabaya dengan periode 1 - 3 Januari 2022 dengan data GFS dan ERA5. Interval waktu dari kedua data tersebut adalah 6 jam, yaitu pukul 00:00, 06:00, 12:00, dan 18:00 waktu Zulu (GMT+0). Langkah awal sebelum menjalankan WRF adalah menentukan lokasi dan waktu terlebih dahulu di dalam program WPS. Seperti yang telah dijelaskan sebelumnya, WPS memiliki 3 program utama, yaitu `geogrid.exe`, `ungrib.exe`, dan `metgrid.exe`.

Sesuai dengan kelanjutan dari subbab sebelumnya mengenai instalasi WRF, Anda saat ini berada di dalam folder `WPS/`. Jika lupa, Anda bisa mengetikkan perintah ini.

```
cd $HOME/WRF-Model/WPS
```

Kemudian, Jika Anda menutup terminal bash atau mematikan laptop/komputer, definisikan kembali ENVIRONMENT VARIABLE seperti pada saat proses memasang WRF dan WPS, hanya LD\_LIBRARY\_PATH dan PATH.

```
export ODIR=$HOME/WRF-Model
export PATH=$ODIR/bin:$PATH
export LD_LIBRARY_PATH=$ODIR/lib:$LD_LIBRARY_PATH
```

Langkah-langkah menjalankan program WPS adalah sebagai berikut.

### 1.4.1.1 Penentuan Lokasi dan Waktu Simulasi

Isi dari file tersebut pada saat awal kurang lebih seperti ini.

```
&share
wrf_core = 'ARW',
max_dom = 2,
start_date = '2019-09-04_12:00:00','2019-09-04_12:00:00',
end_date   = '2019-09-06_00:00:00','2019-09-04_12:00:00',
interval_seconds = 10800
/

&geogrid
parent_id      = 1, 1,
parent_grid_ratio = 1, 3,
i_parent_start = 1, 53,
j_parent_start = 1, 25,
e_we           = 150, 220,
e_sn           = 130, 214,
geog_data_res = 'default','default',
dx = 15000,
dy = 15000,
map_proj = 'lambert',
ref_lat  = 33.00,
ref_lon  = -79.00,
truelat1 = 30.0,
truelat2 = 60.0,
stand_lon = -79.0,
geog_data_path = '/glade/work/wrfhelp/WPS_GEOG/'
/
```

## 1 Dynamical Downscaling: Model Weather Research Forecasting (WRF)

```
&ungrib
  out_format = 'WPS',
  prefix = 'FILE',
/
```

```
&metgrid
  fg_name = 'FILE'
/
```

Terdapat berbagai macam parameter di dalam `&share`, `&geogrid`, `&ungrib`, dan `&metgrid`. Anda perlu mengubah beberapa parameter tersebut yang dapat disesuaikan dengan simulasi. Tapi, Anda harus memperhatikan aturan atau template yang telah diberikan di dalam panduan pengguna. Untuk lebih rincinya, Anda bisa lihat di [Panduan Pengguna WRF-ARW Bab 3](#). Di dalam tahapan ini, Anda perlu mengubah parameter-parameter yang ada di dalam `&geogrid`. Anda dapat mengubah file ini dengan membukanya menggunakan aplikasi apa saja (mis. Notepad, VS Code, Notepad++) atau dapat langsung dibuka di terminal dengan perintah `nano namelist.wps`. Pada contoh yang telah kami berikan, Anda perlu mengganti parameter pada bagian `&share` dan `&geogrid`. Penjelasan setiap variabel dapat dilihat pada Tabel 1.2. File yang telah berubah menjadi seperti di bawan ini.

```
&share
  wrf_core = 'ARW',
  max_dom = 3,
  start_year  = 2022, 2022, 2022,
  start_month = 01,   01,   01,
  start_day   = 01,   01,   01,
  start_hour  = 00,   00,   00,
  end_year    = 2022, 2022, 2022,
  end_month   = 01,   01,   01,
  end_day     = 03,   03,   03,
  end_hour    = 18,   18,   18,
  interval_seconds = 21600,
  io_form_geogrid = 2,
/

&geogrid
  parent_id      = 1, 1, 1,
  parent_grid_ratio = 1, 3, 9,
  i_parent_start  = 1, 11, 15,
  j_parent_start  = 1, 11, 15,
  e_we           = 33, 40, 46,
  e_sn           = 33, 40, 46,
  geog_data_res  = 'default','default','default',
  dx = 18000,
  dy = 18000,
  map_proj = 'mercator',
```



#### 1.4 Prosedur Praktikum: Menjalankan Simulasi WRF-ARW

```

ref_lat   = -7.328457529144323,
ref_lon   = 112.7412453984603,
truelat1  = -7.328457529144323,
geog_data_path = '/home/absen/WRF-Model/data/WPS_GEOG/'
/

&ungrib
  out_format = 'WPS',
  prefix = 'FILE',
/

&metgrid
  fg_name = 'FILE'
  io_form_metgrid = 2,
/

```

Tabel 1.2: Informasi beberapa variabel di dalam **namelist.wps**

Variabel	Keterangan
<b>wrf_core</b>	Tipe penyelesaian WRF, ARW atau NMM
<b>max_dom</b>	Jumlah domain, semakin banyak domain maka semakin tinggi resolusi spasialnya
<b>start_date</b>	Waktu mulai simulasi sesuai dengan data masukan. Sesuai dengan jumlah domain. Format: YYYY-MM-DD_HH:mm:ss
<b>end_date</b>	Waktu akhir simulasi sesuai dengan data masukan. Sesuai dengan jumlah domain. Format: YYYY-MM-DD_HH:mm:ss
<b>interval_seconds</b>	Interval waktu dari data masukan (dalam detik)
<b>io_from_geogrib</b>	Tipe format file luaran <b>geogrid.exe</b> (1 = binary, 2 = netcdf, 3 = GRIB1)
<b>parent_id</b>	Untuk domain paling kasar, nilainya 1. Domain selanjutnya juga bernilai 1 yang menandakan bahwa subdomain merupakan bagian dari domain utamanya.
<b>parent_grid_ratio</b>	Rasio piksel dari dx dan dy pada domain ke-1. Domain paling kasar adalah 1 dan selanjutnya mengikuti rasio yang diinginkan.
<b>i_parent_start</b>	Nomor indeks acuan untuk koordinat X. Domain paling kasar bernilai 1, ditentukan dari kiri-bawah
<b>j_parent_start</b>	Nomor indeks acuan untuk koordinat Y. Domain paling kasar bernilai 1, ditentukan dari kiri-bawah
<b>e_we</b>	Jumlah grid/piksel dari barat ke timur mengikuti rumus <b>parent_grid_ratio * N + 1</b> , dengan $N > 0$
<b>e_sn</b>	Jumlah grid/piksel dari selatan ke utara mengikuti rumus <b>parent_grid_ratio * N + 1</b> , dengan $N > 0$
<b>geog_data_ref</b>	Pemilihan resolusi spasial dari data statik
<b>dx</b>	Resolusi spasial pada koordinat X (dalam meter)
<b>dy</b>	Resolusi spasial pada koordinat Y (dalam meter)
<b>map_proj</b>	Sistem proyeksi peta ( <b>mercator</b> , <b>lambert</b> , <b>polar</b> , <b>lat-lon</b> ). Untuk simulasi di sekitar khatulistiwa, direkomendasikan menggunakan <b>mercator</b>
<b>ref_lat</b>	Koordinat lintang titik tengah acuan dari domain paling kasar

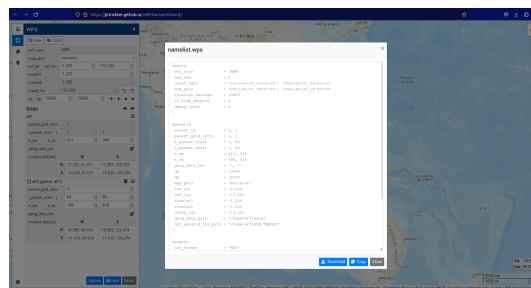
## 1 Dynamical Downscaling: Model Weather Research Forecasting (WRF)

Variabel	Keterangan
ref_lon	Koordinat bujur titik tengah acuan dari domain paling kasar
truelat1	Koordinat lintang sebenarnya. Dibutuhkan untuk sistem proyeksi mercator, polar, dan lambert
geog_data_path	Lokasi folder WPS_GEOG

Pada parameter `start_date` dan `end_date`, Anda bisa menggunakan parameter lain dengan membagi masing-masing format tahun (`*_year`), bulan (`*_month`), tanggal (`*_day`), dan jam (`*_hour`).

Berkaitan dengan skala resolusi spasial pada simulasi yang akan dijalankan dengan pemilihan lokasi di Kota Surabaya, skala tertingginya adalah 2 km. Anda perhatikan parameter `parent_grid_ratio`. Nilai 1, 3, dan 9 secara berturut-turut merupakan rasio terhadap `dx` atau `dy` untuk setiap domain. Nilai 1 berarti untuk domain terluar dengan skala  $1/1 * 18000 = 18000$  meter, sedangkan nilai 3 untuk domain ke-2 dengan skala  $1/3 * 18000 = 6000$  meter.

Pada bagian `&geogrid`, penentuan lokasi kajian ini cukup rumit. Anda bisa menggunakan halaman Github <https://jiririchter.github.io/WRFDomainWizard> untuk membuat file `namelist.wps` sesuai dengan lokasi yang Anda inginkan, setidaknya Anda dapat menyalin teks pada bagian `&geogrid` Gambar 1.12.



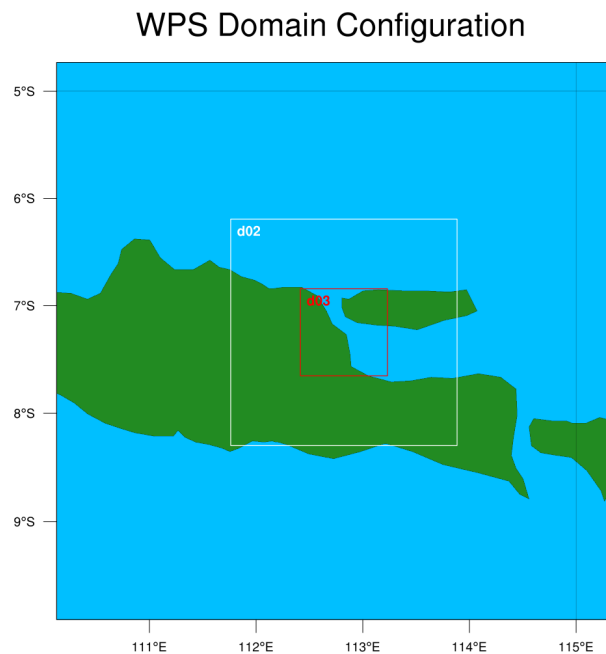
Gambar 1.12: WRF Domain Wizard oleh Jiririchter

Untuk memverifikasi kesesuaian pemilihan lokasi yang telah dibuat, Anda dapat menggunakan skrip `plotgrids_new.ncl` di dalam folder `util/`. Bukalah file tersebut. Cari dan ubahlah variabel `type = x11` menjadi `type = png` untuk mengatur luaran file dalam bentuk gambar (format `.png`). Untuk menjalankan skrip ini, aktifkan terlebih dahulu `environment ncl` dengan perintah `source activate ncl` atau `conda activate ncl`. Kemudian, jalankan perintah berikut ini. File `.png` akan muncul di dalam folder `WPS/` dengan nama `wps_show_dom.png`. Hasilnya seperti Gambar 1.13

```
ncl util/plotgrids_new.ncl
```

### 1.4.1.2 Menjalankan Program geogrid.exe

Selanjutnya, Anda dapat menjalankan program `geogrid.exe` dengan perintah di bawah ini. Hasilnya, terdapat 3 file dengan nama `geo_em.d0x.nc` ( $x = \text{nomor domain; } 1, 2, 3$ ) karena pengaturan `max_dom = 3`.



Gambar 1.13: Konfigurasi domain WRF

```
./geogrid.exe
```

```
ls geo_em*
```

```
geo_em.d01.nc  geo_em.d02.nc  geo_em.d03.nc
```

#### 1.4.1.3 Menyambungkan File Data Masukan dan Tabel Variabel

WPS menyediakan program `link_grib.csh` untuk menyambungkan file data masukan ke dalam folder `WPS/` dengan membuat shortcut yang bernama `GRIBFILE.*` (`GRIBFILE.AAA`, `GRIBFILE.AAB`, ...). Perintahnya sebagai berikut.

```
./link_grib.csh $ODIR/data/GFS/*
```

Kemudian, buatlah shortcut dengan nama `Vtable` di dalam folder `WPS/` dari file yang berada di dalam folder `ungrib/Variable_Tables`. Untuk data GFS, nama file tersebut adalah `Vtable.GFS`, sedangkan ERA5 adalah `Vtable.ERA-Interim.pl`. Perintahnya sebagai berikut.

```
ln -sf ungrib/Variable_Tables/Vtable.GFS Vtable # GFS
ln -sf ungrib/Variable_Tables/Vtable.ERA-Interim.pl Vtable # ERA5
```

#### 1.4.1.4 Menjalankan Program ungrib.exe

Setelah menyambungkan data masukan serta tabel variabel, jalankan program **ungrib.exe** dengan perintah berikut.

```
./ungrib.exe
```

Luaran dari program ini adalah file dengan nama **FILE:\*** yang memiliki format **.nc** (netcdf).

#### 1.4.1.5 Menjalankan Program metgrid.exe

Jalankan program **metgrid.exe** dengan perintah berikut.

```
./metgrid.exe
```

Luaran dari program ini adalah file dengan nama **met\_em\*** yang memiliki format **.nc**. File-file ini nanti yang akan dipindahkan menuju folder **WRF/test/em\_real** atau Anda dapat juga membuat shortcut. Perintahnya sebagai berikut.

```
mv met_em* $ODIR/WRF/test/em_real
# atau
ln -sf met_em* $ODIR/WRF/test/em_real
```

### 1.4.2 Program WRF

Pada bagian ini, program WRF yang digunakan hanya 2: **real.exe** dan **wrf.exe**. Sebelum itu, Anda diharuskan mengubah beberapa parameter pada file **namelist.input** seperti di dalam file **namelist.wps**. Parameter di dalam **namelist.input** sangat banyak karena terdapat bagian pemilihan lokasi dan waktu (disesuaikan dengan **namelist.wps**), pemilihan skema parameter fisik, dan parameter dinamik. Anda dapat membaca lebih lanjut di [Panduan Pengguna WRF Bab 5](#).

### 1.4.2.1 Mengubah isi `namelist.input`

Saat ini Anda masih berada di folder `WPS/`. Buka terlebih dahulu folder `em_real` dengan perintah

```
cd $ODIR/WRF/test/em_real
```

Di dalam folder `em_real`, terdapat 4 program, yaitu `real.exe`, `ndown.exe`, `tc.exe`, dan `wrf.exe`. Nilai `max_dom` pada `namelist.wps` adalah 3. Jika pemilihan lokasi Anda memiliki >1 domain, WRF akan melakukan proses simulasi sampai pada domain tertinggi atau disebut dengan *nesting*. Proses ini juga dibagi lagi menjadi 2, yaitu *nesting* dua arah dan satu arah. Untuk *nesting* satu arah, program yang digunakan bisa atau tanpa dengan program `ndown.exe`. Penggunaan *nesting* satu arah biasanya pengguna hanya tertarik pada analisis domain tertinggi. Penggunaan `ndown.exe` juga sering digunakan pada turunan WRF, yaitu WRF-Chem jika terdapat >1 domain karena sampai modul ini dibuat, belum ada pembaruan terkait algoritma *downscaling* pada WRF-Chem dengan *nesting* dua arah.

Bukalah file `namelist.input` dengan cara sama seperti Anda membuka `namelist.wps`. Ketika pertama kali dibuka, file `namelist.input` seperti di bawah ini. Anda juga dapat melihat beberapa contoh/template lain (mis. `namelist.input.4km`, `namelist.input.chem`, `namelist.input.volc`, ...) sesuai dengan kebutuhan tertentu.

```
&time_control
run_days              = 0,
run_hours             = 36,
run_minutes           = 0,
run_seconds           = 0,
start_year            = 2019, 2019,
start_month           = 09,  09,
start_day             = 04,  04,
start_hour            = 12,  12,
end_year              = 2019, 2019,
end_month             = 09,  09,
end_day               = 06,  06,
end_hour              = 00,  00,
interval_seconds      = 10800
input_from_file       = .true.,.true.,
history_interval      = 60,  60,
frames_per_outfile    = 1,  1,
restart               = .false.,
restart_interval      = 7200,
io_form_history        = 2
io_form_restart       = 2
io_form_input         = 2
io_form_boundary      = 2
/
```

# 1 Dynamical Downscaling: Model Weather Research Forecasting (WRF)

```

&domains
time_step                = 90,
time_step_fract_num      = 0,
time_step_fract_den      = 1,
max_dom                  = 2,
e_we                     = 150,    220,
e_sn                     = 130,    214,
e_vert                   = 45,     45,
dzstretch_s              = 1.1
p_top_requested           = 5000,
num_metgrid_levels       = 34,
num_metgrid_soil_levels  = 4,
dx                       = 15000,
dy                       = 15000,
grid_id                  = 1,      2,
parent_id                = 0,      1,
i_parent_start           = 1,      53,
j_parent_start           = 1,      25,
parent_grid_ratio         = 1,      3,
parent_time_step_ratio   = 1,      3,
feedback                 = 1,
smooth_option            = 0
/

&physics
physics_suite            = 'CONUS'
mp_physics                = -1,    -1,
cu_physics                = -1,    -1,
ra_lw_physics             = -1,    -1,
ra_sw_physics             = -1,    -1,
bl_pbl_physics            = -1,    -1,
sf_sfclay_physics         = -1,    -1,
sf_surface_physics        = -1,    -1,
radt                      = 15,    15,
bldt                      = 0,      0,
cudt                      = 0,      0,
icloud                   = 1,
num_land_cat              = 21,
sf_urban_physics          = 0,      0,
fractional_seaice         = 1,
/

&fdda
/

&dynamics

```

#### 1.4 Prosedur Praktikum: Menjalankan Simulasi WRF-ARW

```
hybrid_opt          = 2,
w_damping           = 0,
diff_opt            = 2,          2,
km_opt              = 4,          4,
diff_6th_opt        = 0,          0,
diff_6th_factor     = 0.12,      0.12,
base_temp           = 290.
damp_opt            = 3,
zdamp               = 5000.,      5000.,
dampcoef            = 0.2,        0.2,
khdif               = 0,          0,
kvdif               = 0,          0,
non_hydrostatic     = .true.,     .true.,
moist_adv_opt        = 1,          1,
scalar_adv_opt       = 1,          1,
gwd_opt             = 1,          0,
/

&bdy_control
spec_bdy_width      = 5,
specified            = .true.
/

&grib2
/

&namelist_quilt
nio_tasks_per_group = 0,
nio_groups = 1,
/
```

Pada file ini, samakan beberapa parameter seperti di file `namelist.wps`. Perhatikan Tabel 1.3. Anda cukup mencari parameter yang sama antara `namelist.wps` dengan `namelist.input`, tetapi tidak semuanya ada di dalam `namelist.input`. Untuk parameter lainnya, seperti `parent_time_step_ratio`, `time_step`, `history_interval`, `frame_per_outfile`, dan seterusnya, Anda dapat membacanya lebih banyak di [Panduan Pengguna WRF Bab 5](#) atau bisa dilihat pada file `README.namelist` di dalam folder `test/em_real` untuk setiap penjelasan singkat berbagai parameter.

Bagian `&time_control` berfungsi sebagai pengaturan waktu simulasi serta luaran yang akan dihasilkan. Sebagai informasi, file luaran WRF berformat NetCDF (.nc) dengan nama `wrfout_<domain>_<yyyy>-<mm>-<dd>_<HH>` dimana

- domain: identitas domain (d01, d02, ...)
- yyyy: tahun, dengan format 4 digit
- mm: bulan, dengan format 2 digit
- dd: tanggal, dengan format 2 digit

## 1 Dynamical Downscaling: Model Weather Research Forecasting (WRF)

- HH: jam, dengan format 2 digit
- MM: menit, dengan format 2 digit
- SS: detik, dengan format 2 digit

Pada parameter `run_days`, `run_hours`, `run_minutes`, dan `run_seconds`, ini dapat dihitung dari selisih waktu akhir simulasi dengan awal simulasi. Pada simulasi yang akan dicoba dalam modul ini, yaitu 1 Januari 2022 pukul 00:00 UTC hingga 3 Januari 2022 18:00 UTC, nilai dari `run_days` dan `run_hours` secara berturut-turut adalah 2 dan 18. Anda juga dapat mengatur `run_days` ini menjadi 0 setelah dikonversi menjadi jam (2 hari = 48 jam) dan tambahkan ke `run_hours`, yaitu menjadi 66. Parameter `history_interval` digunakan untuk meletakkan nilai pada file luaran WRF dalam format `.nc` dengan waktu tertentu (dalam menit). Misalkan diatur ke 60, berarti hasil perhitungan dari berbagai algoritma WRF dimasukkan ke file setiap 60 menit sekali. Anda bebas mengatur angka pada parameter ini. Dampaknya, ukuran file akan semakin besar jika Anda mengatur nilainya kecil. Tentu ini tidak akan menjadi masalah apabila ruang kosong penyimpanan internal/eksternal Anda masih banyak. Banyaknya file luaran dapat pula diatur jumlahnya, yaitu di dalam parameter `frames_per_outfile`. Jika parameter diatur pada `frames_per_outfile` = 1, `history_interval` = 60, dan `run_hours` = 66, berarti file luaran yang akan dibuat dan disimpan ke dalam penyimpanan adalah sebanyak 66 file. Agar lebih efektif saat akan melakukan analisis, kami menyarankan untuk mengatur `frames_per_outfile` = 1000.

Kemudian untuk bagian `&domains`, digunakan untuk mengatur kondisi dari domain agar dapat sesuai dengan data masukan (banyak grid horizontal dan vertikal, posisi, rasio grid, tipe interpolasi). Pada parameter `feedback`, Anda dapat mengatur tipe *nesting* dua arah (1) atau satu arah (0). Parameter `num_metgrid_levels` dan `num_metgrid_soil_levels` harus diatur sesuai dengan yang ada di dalam salah satu file `met_em*`. Untuk melihatnya, gunakan perintah di bawah ini. Nilai kedua parameter `num_metgrid_levels` dapat berbeda sesuai dengan data masukan yang digunakan. Sebagai contoh, GFS memiliki `num_metgrid_levels` = 34, sedangkan ERA5 `num_metgrid_levels` = 38.

```
ncdump -h met_em.d01.2022-01-01_00:00:00.nc | grep num_metgrid_levels      # = 34
ncdump -h met_em.d01.2022-01-01_00:00:00.nc | grep NUM_METGRID_SOIL_LEVELS # = 4
```

Di dalam bagian `&physics`, terdapat berbagai skema parameterisasi dalam penyelesaian perhitungan pembentukan awan, skemar radiasi, lapisan perbatas, serta proses-proses di permukaan tanah. Pada parameter `physics_suite`, Anda bisa mengaturnya ke `TROPICAL` karena wilayah yang ingin disimulasikan berada di daerah tropis. Ketika Anda mendefinisikan `physics_suite` = `'TROPICAL'`, Anda tidak perlu lagi menambahkan angka pada parameter di bawah ini.

- `mp_physics` = 6: WSM6
- `cu_physics` = 16: New-Tiedke
- `ra_lw_physics` = 4: Rapid Radiative Model Transformation for GCM (RRTMG)
- `ra_sw_physics` = 4: Rapid Radiative Model Transformation for GCM
- `bl_pbl_physics` = 1: Yonsei University
- `sf_sfclay_physics` = 91: MM5 Monin-Obukhov
- `sf_surface_physics` = 2: Noah Land Surface Model

Anda hanya perlu menambahkan nilai -1 pada parameter-parameter tersebut, tentu saja sesuai dengan banyaknya domain. Anda juga dapat menghilangkan parameter `physics_suite` dan mengganti



parameter `mp_physics`, `cu_physics`, `sf_surface_physics`, `sf_sfclay_physics`, `ra_lw_physics`, `ra_sw_physics`, dan `bl_pbl_physics` ke opsi lain yang ada di dalam [Panduan Pengguna WRF Bab 5](#). Anda harus berhati-hati dan memperhatikan pemilihan skema parameterisasi karena terdapat parameter yang harus ditambahkan di dalam bagian `&physics`, menambahkan bagian lain, atau kombinasi antar skema. Sebagai contoh untuk `sf_surface_physics = 4` (*Noah-MP Land Surface Model*), perlu menambahkan bagian `&noah_mp`; Ketika `cu_physics = 14` (*Scale-aware SAS*), perlu menambahkan parameter `shcu_physics = 4`; Parameter `bl_pbl_physics = 2` (*Mellor-Yamada-Janjic, MYJ*) nilai dari parameter `sf_sfclay_physics = 2` (*Eta Model*). Untuk pengaturan nilai-nilai di dalam `&physics`, Anda harus menambahkannya sebanyak jumlah domain. Anda juga dapat menonaktifkan skema parameterisasi tertentu dengan mengubahnya ke 0, misalnya pada `cu_physics` untuk domain  $<10$  km.

Tabel 1.3: Informasi sebagian variabel di dalam `namelist.input` yang perlu disesuaikan dengan `namelist.wps`

Parameter di <code>namelist.wps</code>	Parameter di <code>namelist.input</code>
<code>start_year = 2022, 2022, 2022,</code>	<code>start_year = 2022, 2022, 2022,</code>
<code>start_month = 01, 01, 01,</code>	<code>start_month = 01, 01, 01,</code>
<code>start_day = 01, 01, 01,</code>	<code>start_day = 01, 01, 01,</code>
<code>start_hour = 00, 00, 00,</code>	<code>start_hour = 00, 00, 00,</code>
<code>end_year = 2022, 2022, 2022,</code>	<code>end_year = 2022, 2022, 2022,</code>
<code>end_month = 01, 01, 01,</code>	<code>end_month = 01, 01, 01,</code>
<code>end_day = 03, 03, 03,</code>	<code>end_day = 03, 03, 03,</code>
<code>end_hour = 18, 18, 18,</code>	<code>end_hour = 18, 18, 18,</code>
-	<code>run_days = 2,</code>
-	<code>run_hour = 18,</code>
<code>max_dom = 3,</code>	<code>max_dom = 3,</code>
<code>interval_seconds = 21600,</code>	<code>interval_seconds = 21600,</code>
<code>parent_id = 1, 1, 1,</code>	<code>parent_id = 1, 1, 1,</code>
<code>parent_grid_ratio = 1,</code>	<code>parent_grid_ratio = 1, 3, 9,</code>
<code>3, 9,</code>	
<code>dx = 15000,</code>	<code>dx = 15000,</code>
<code>dy = 15000,</code>	<code>dy = 15000,</code>
<code>i_parent_start = 1, 11,</code>	<code>i_parent_start = 1, 11, 15,</code>
<code>15,</code>	
<code>j_parent_start = 1, 11,</code>	<code>j_parent_start = 1, 11, 15,</code>
<code>15,</code>	
<code>e_we = 33, 40, 46,</code>	<code>e_we = 33, 40, 46,</code>
<code>e_sn = 33, 40, 46,</code>	<code>e_sn = 33, 40, 46,</code>
-	<code>parent_time_step_ratio = 1, 3, 9, (sama dengan</code>
-	<code>parent_grid_ratio)</code>
-	<code>time_step = 90, dengan rumus <math>6 * dx</math> (dalam km)</code>

Isi file `namelist.input` yang telah diubah sepenuhnya menjadi seperti ini.

```

&time_control
  run_days           = 2,
  run_hours          = 18,
  run_minutes        = 0,
  run_seconds        = 0,
  start_year         = 2022, 2022, 2022,
  start_month        = 01, 01, 01,
  start_day          = 01, 01, 01,
  start_hour         = 00, 00, 00,
  end_year           = 2022, 2022, 2022,
  end_month          = 01, 01, 01,
  end_day            = 03, 03, 03,
  end_hour           = 18, 18, 18,
  interval_seconds   = 21600,
  input_from_file    = .true., .true., .true.,
  history_interval   = 60, 60, 60,
  frames_per_outfile = 1000, 1000, 1000,
  restart            = .false.,
  restart_interval   = 7200,
  io_form_history     = 2
  io_form_restart    = 2
  io_form_input       = 2
  io_form_boundary   = 2
/

&domains
  time_step          = 90,
  time_step_fract_num = 0,
  time_step_fract_den = 1,
  max_dom            = 3,
  e_we               = 33, 40, 46,
  e_sn               = 33, 40, 46,
  e_vert             = 44, 44, 44,
  dzstretch_s        = 1.1
  p_top_requested    = 5000,
  use_surface        = .false.,
  sfcf_to_sfcf       = .true.,
  num_metgrid_levels = 34,
  num_metgrid_soil_levels = 4,
  dx                 = 18000,
  dy                 = 18000,
  grid_id            = 1, 2, 3,
  parent_id          = 1, 1, 1,
  i_parent_start     = 1, 11, 15,
  j_parent_start     = 1, 11, 15,
  parent_grid_ratio   = 1, 3, 9,

```

#### 1.4 Prosedur Praktikum: Menjalankan Simulasi WRF-ARW

```
parent_time_step_ratio      = 1,      3,      9,
feedback                    = 1,
smooth_option               = 0
/

&physics
physics_suite               = 'TROPICAL'
mp_physics                  = -1,      -1,      -1,
cu_physics                  = -1,      0,      0,
ra_lw_physics               = -1,      -1,      -1,
ra_sw_physics               = -1,      -1,      -1,
bl_pbl_physics              = -1,      -1,      -1,
sf_sfclay_physics           = -1,      -1,      -1,
sf_surface_physics          = -1,      -1,      -1,
radt                        = 18,      18,      18,
bldt                        = 0,      0,      0,
cudt                        = 0,      0,      0,
icloud                      = 1,
num_land_cat                = 21,
sf_urban_physics            = 0,      0,      0,
fractional_seaice           = 1,
/

&fdda
/

&dynamics
hybrid_opt                  = 2,
w_damping                   = 0,
diff_opt                    = 2,      2,      2,
km_opt                      = 4,      4,      4,
diff_6th_opt                = 0,      0,      0,
diff_6th_factor             = 0.12,    0.12, 0.12,
base_temp                   = 290.
damp_opt                    = 3,
zdamp                       = 5000.,   5000., 5000.,
dampcoef                    = 0.2,     0.2,     0.2,
khdif                       = 0,      0,      0,
kvdif                       = 0,      0,      0,
non_hydrostatic             = .true.,   .true., .true.,
moist_adv_opt               = 1,      1,      1,
scalar_adv_opt              = 1,      1,      1,
gwd_opt                     = 1,      0,      0,
/

&bdy_control
```

## 1 Dynamical Downscaling: Model Weather Research Forecasting (WRF)

```
spec_bdy_width          = 5,  
specified                = .true.  
/  
  
&grib2  
/  
  
&namelist_quilt  
nio_tasks_per_group = 0,  
nio_groups = 1,  
/  

```

### 1.4.2.2 Menjalankan program `real.exe` dan `wrf.exe`

Program `real.exe` mengeluarkan file-file dalam format `.nc`, yaitu `wrfbdy_d01` dan `wrfinput_<domain>`. Perintah menjalankan program ini adalah sebagai berikut.

```
./real.exe
```

Untuk melihat respon dari program ini, Anda dapat melihat file `rsl.error.0000` dengan perintah

```
tail rsl.error.0000 -n 1
```

Jika respon yang diberikan terdapat kalimat `real_em: SUCCESS COMPLETE REAL_EM INIT`, Anda bisa melanjutkan ke tahapan `wrf.exe`.

```
./wrf.exe
```

Program `wrf.exe` dikatakan berhasil apabila terdapat kalimat `wrf: SUCCESS COMPLETE WRF` pada baris terakhir `rsl.error.0000`. Langkah selanjutnya, Anda dapat menganalisis luaran WRF dengan aplikasi apapun yang Anda bisa, selama mendukung format `*.nc`. File simulasi luaran WRF bernama `wrfout_<domain>_<yyyy>-<mm>-<dd>_<HH>:<MM>:<SS>`. Untuk simulasi ini, terdapat 3 file `wrfout_d0*`. Sebenarnya, file-file tersebut berformat `*.nc` walaupun tidak tertera pada nama file. File `wrfout_d0*` memiliki interval waktu per 1 jam, dari pukul 00:00 UTC 1 Januari - 18:00 UTC 3 Januari 2022.

### 1.4.2.3 Tutup terminal Bash Anda

## 1.5 Prosedur Praktikum: Analisis Luaran WRF

Terdapat 3 file luaran WRF yang telah Anda jalankan dengan masing-masing memiliki resolusi spasial yang berbeda, yaitu 18 km, 6 km, dan 2 km dengan pemilihan lokasi titik tengah longitude dan latitude di Kota Surabaya. Selanjutnya, Anda dapat melihat hasil dari simulasi tersebut dengan berbagai macam

perangkat lunak, seperti ArcMap, Matlab, NCL, R, atau Python. Untuk praktikum ini, kami memberikan contoh dengan menggunakan Python di dalam *Environment ncl* yang telah Anda pasang sebelumnya.

### 1.5.1 Membuka Jupyter Notebook

Anda bisa menggunakan kode editor ini atau aplikasi lainnya, seperti Visual Studio Code, Notepad++, atau Atom. Untuk langkah-langkah di bawah ini, kami lebih menjelaskan pada **Jupyter Notebook**. Kode editor ini diketik melalui aplikasi Browser default Anda, misalnya Google Chrome, Mozilla Firefox, atau Microsoft Edge. Ikuti langkah-langkah berikut ini.

1. Bukalah terminal Bash Anda. Saat ini, Anda berada di direktori `$HOME` atau `~`.
2. Bukalah direktori `WRF-Model` dengan mengetik perintah

```
cd WRF-Model
```

3. Untuk memulai pengetikan kode Python, aktifkan terlebih dahulu *Environment ncl* dengan perintah `conda activate ncl`.
4. Pengetikan kode Python dapat Anda langsung terapkan pada mode interaktif dengan mengetik perintah `python` pada terminal. Kami menyarankan Anda untuk menggunakan **Jupyter Notebook** agar hasil kode langsung bisa ditampilkan di bawahnya. Pasang **Jupyter Notebook** pada terminal dengan mengetik perintah

```
mamba install -c conda-forge jupyter notebook
```

5. Ketikkan perintah berikut ini untuk memulai pengetikan kode Python. Browser default Anda akan terbuka dengan menampilkan kode editor **Jupyter Notebook** (Gambar 1.14)

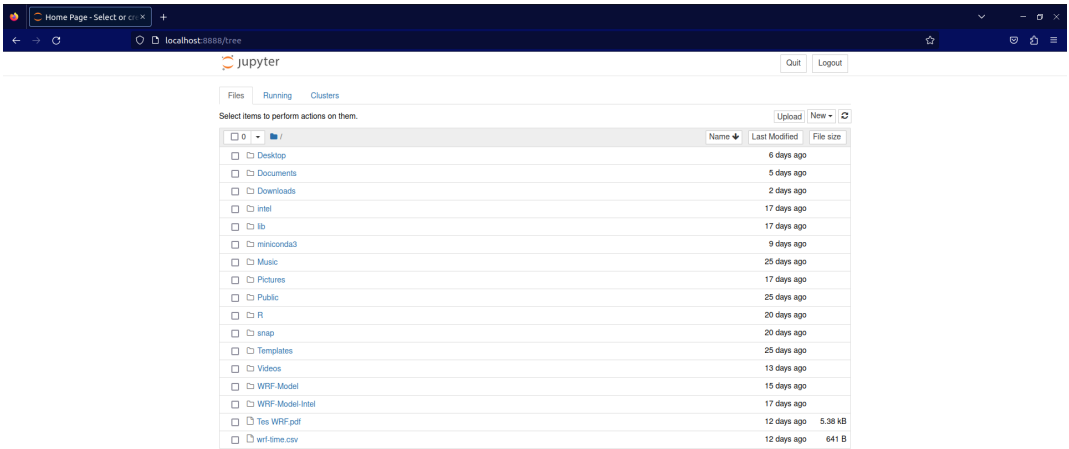
```
jupyter notebook
```

6. Klik **New** dan pilih **Python 3 (ipykernel)**. Tampilan awal **Jupyter Notebook** seperti pada Gambar 1.15
7. Anda dapat langsung mendapatkan hasil dari kode yang ditulis pada sel (lihat `In [1]`) (Gambar 1.16)

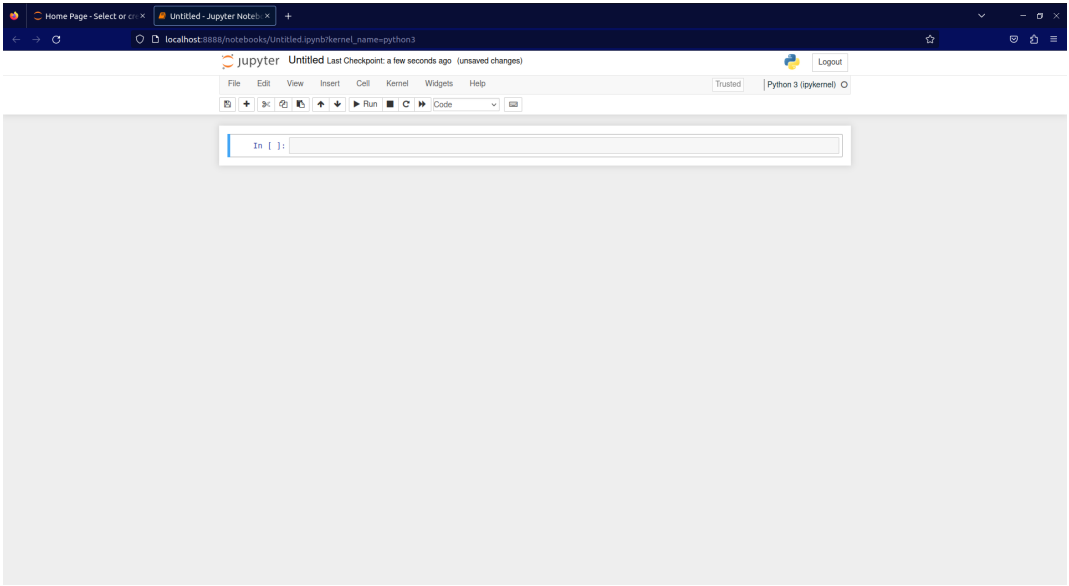
### 1.5.2 Menggunakan `wrf-python`

Package `wrf-python` [5] dikembangkan oleh NCAR yang dikhususkan untuk *Post-Processing* luaran WRF. Package ini mendukung pembacaan file, perhitungan interpolasi, serta visualisasi WRF. Perhatikan langkah-langkah berikut cara penggunaannya mulai dari pembacaan file hingga pembuatan grafik. Package berikut ini dibutuhkan untuk melakukan langkah-langkah tersebut.

1 Dynamical Downscaling: *Model Weather Research Forecasting (WRF)*

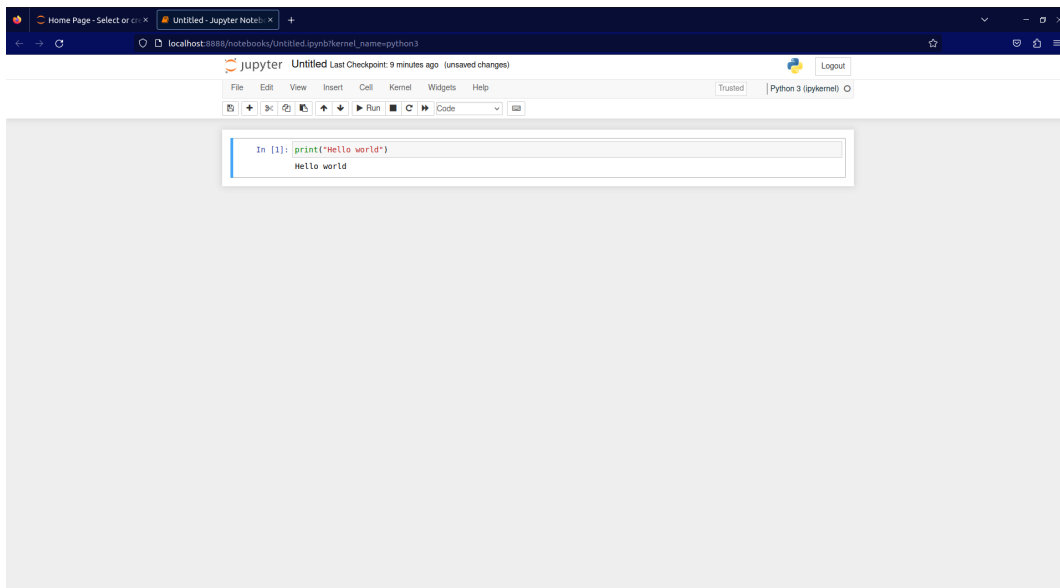


Gambar 1.14: Tampilan Jupyter Notebook



Gambar 1.15: Tampilan awal Jupyter Notebook

## 1.5 Prosedur Praktikum: Analisis Luaran WRF



Gambar 1.16: Tampilan interaktif Jupyter Notebook

```
from netCDF4 import Dataset
import wrf
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.cm import get_cmap
import cartopy.crs as crs
from cartopy.feature import NaturalEarthFeature
```

### 1.5.2.1 Mengekstrak Variabel WRF

1. Saat ini, Anda berada di direktori `~/WRF-Model`. File luaran WRF berada di folder `~/WRF-Model/WRF/test/em_real`. Buat variabel yang berisi lokasi WRF Anda

```
wrf_path = '/home/absen/WRF-Model/WRF/test/em_real'
```

2. Buka salah satu file `wrfout_d0*`, misalnya `wrfout_d03_2022-01-01_00:00:00` dan cetak variabel tersebut, misalkan variabel `T2` (*Air Temperature at 2m*: suhu udara 2 meter dari permukaan tanah). Isinya adalah beberapa metadata. Anda dapat mengetahui variabel-variabel di dalamnya dengan menambahkan metode `.variables`. Anda dapat melihat penjelasan variabel-variabel pada panduan pengguna WRF-ARW.

```
wrf_d03 = Dataset(f"{wrf_path}/wrfout_d03_2022-01-01_00:00:00")
```

## 1 Dynamical Downscaling: *Model Weather Research Forecasting (WRF)*

3. Anda dapat mengambil variabel dengan fungsi `wrf.getvar()` dengan menyertakan argumen dari nama variabel WRF. Untuk mengambil T2, gunakan perintah di bawah ini. Perhatikan hasil yang diperoleh merupakan tipe `xarray.DataArray` dan metode `wrf.ALL_TIMES` pada argumen `timeidx` berguna untuk mengambil variabel dengan seluruh waktu, mulai dari awal hingga akhir simulasi.

```
wrf_t2 = wrf.getvar(wrf_d03, "T2", timeidx=wrf.ALL_TIMES)
```

4. Untuk mendapatkan nilai koordinat latitude dan longitude dari variabel T2, Anda dapat memakai fungsi `wrf.latlon_coords()`. Pastikan dalam satu baris terdapat 2 variabel.

```
lats, lons = wrf.latlon_coords(wrf_t2)
```

5. Untuk mendapatkan waktu, Anda dapat memanggil *Coordinates Time* yang telah tersedia di dalam `wrf_t2`.

```
wrf_time = wrf_t2.Time
```

6. Sebagai tambahan, Anda juga dapat mengekstrak beberapa variabel yang tidak tersedia di dalam WRF, tentunya terbatas, seperti resultan kecepatan angin (`wspd`), arah angin (`wdir`), *Convective Available Potential Energy* (CAPE; `cape3d_only/mcape`), atau suhu titik embun (`td/td2`). Anda dapat membaca lebih lanjut di <https://wrf-python.readthedocs.io/en/latest/diagnostics.html>.
7. Variabel curah hujan tidak ada di dalam WRF. Anda harus mengekstrak dan menjumlahkan variabel `RAINNC` (*Accumulated Total Cumulus Precipitation*) dan `RAINNC` (*Accumulated Total Grid Scale Cumulus Precipitation*), maka diperoleh akumulasi curah hujan dari awal sampai akhir waktu simulasi. Oleh karena akumulasi, Anda perlu mengurangi curah hujan dari waktu ke  $t$  dengan  $t - 1$ . Pada Python, gunakan algoritma perulangan.

```
# Ekstrak RAINC dan RAINNC
rainc = wrf.getvar(wrf_d03, "RAINC" , timeidx = wrf.ALL_TIMES)
rainnc = wrf.getvar(wrf_d03, "RAINNC", timeidx = wrf.ALL_TIMES)

# Menghitung Curah hujan akumulasi
rain = rainc + rainnc
rain_diff = rain.copy() # Metode .copy() agar var rain tidak ikut berubah

# Lakukan perulangan
length = len(wrf_time.values)
for i in range(1, length):
    rain_diff[i, :, :] = rain[i, :, :] - rain[i-1, :, :]

# Copy Attribute (dari RAINC atau RAINNC)
rain_diff.attrs = rainc.attrs
# Tambahkan deskripsi
```



```
rain_diff.attrs["description"] = "Total Rainfall"
```

### 1.5.2.2 Membuat Grafik WRF

Contoh pembuatan grafik di bawah ini adalah grafik secara spasial dan temporal, serta vertikal.

#### 1.5.2.2.1 Spasial

Setelah berhasil mengekstrak variabel dari langkah sebelumnya, langkah-langkah berikut ini adalah pembuatan grafik spasial dari variabel T2. Untuk membuatnya, Anda membutuhkan tambahan package `matplotlib` dan `cartopy`.

1. Anda telah mendefinisikan lokasi (variabel `lats` dan `lons`) pada langkah sebelumnya. Kedua variabel tersebut digunakan dalam membuat grafik spasial.
2. Variabel suhu udara yang telah diekstrak pada langkah sebelumnya memiliki waktu dari awal hingga akhir simulasi. Anda hanya dapat memilih salah satu waktu dalam 1 grafik. Anda juga dapat membuat grafik untuk masing-masing waktu dalam satu grafik dalam bentuk *panel plot*. Untuk membuat grafik pada waktu tertentu, Anda perlu memilih waktu dan mengetahui letak indeksinya.

```
time      = "2022-01-02 13:00:00"
time      = np.array([time], dtype='datetime64[ns]')
time_idx  = np.where(wrf_time.values == time)[0]
time_idx  = int(time_idx)
```

3. Sebagai contoh skrip di bawah ini untuk membuat grafik spasial hanya satu waktu.

```
# Get the cartopy mapping object
cart_proj = wrf.get_cartopy(wrf_t2, timeidx=time_idx)

# Create a figure
fig = plt.figure(figsize=(12,10))
# Set the GeoAxes to the projection used by WRF
ax = plt.axes(projection=cart_proj)

# Add features to axes object
ax.coastlines(linewidth=0.8)

# Activate gridlines
gl = ax.gridlines(draw_labels=True, linewidth=1, color='gray', alpha=0.5,
    ↪ linestyle='--')
gl.top_labels = False
gl.right_labels = False
```

## 1 Dynamical Downscaling: Model Weather Research Forecasting (WRF)

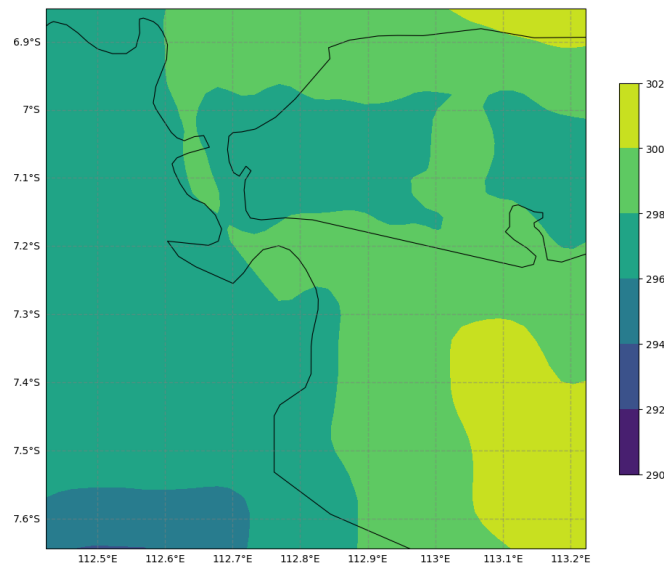
```
# Add features to the map like contour lines
# Set contour levels
lvl = np.arange(290, 304, 2)

# Add filled contour lines
plt.contourf(lons, lats, smooth_t2[time_idx, :, :],
             levels=lvl,
             transform=crs.PlateCarree(),
             cmap='viridis')

# Add colorbar
plt.colorbar(ax=ax, shrink=0.7)

# Manually set extent of map projection
# ax.set_extent([])

plt.show()
```



Gambar 1.17: Grafik suhu udara 2m pada 2022-01-02 13:00:00 UTC

4.

## Daftar Pustaka

- [1] Michael Glotter et al. “Evaluating the utility of dynamical downscaling in agricultural impacts projections”. In: *Proceedings of the National Academy of Sciences* 111.24 (2014), pp. 8776–8781.
- [2] Aneesh Goly, Ramesh SV Teegavarapu, and Arpita Mondal. “Development and evaluation of statistical downscaling models for monthly precipitation”. In: *Earth Interactions* 18.18 (2014), pp. 1–28.
- [3] R Huth and J Kysely. “Constructing site-specific climate change scenarios on a monthly scale using statistical downscaling”. In: *Theoretical and Applied Climatology* 66.1 (2000), pp. 13–27.
- [4] JW Kim et al. “The statistical problem of climate inversion: Determination of the relationship between local and large-scale climate”. In: *Monthly weather review* 112.10 (1984), pp. 2069–2077.
- [5] W Ladwig. *wrf-python Version 1.3.4*. 2017. DOI: <https://doi.org/10.5065/D6W094P1>.
- [6] HA Pahlavan et al. “Improvement of multiple linear regression method for statistical downscaling of monthly precipitation”. In: *International journal of environmental science and technology* 15.9 (2018), pp. 1897–1912.
- [7] C Piani, JO Haerter, and E Coppola. “Statistical bias correction for daily precipitation in regional climate models over Europe”. In: *Theoretical and Applied Climatology* 99.1-2 (2010), pp. 187–192.
- [8] C Piani et al. “Statistical bias correction of global simulated daily precipitation and temperature for the application of hydrological models”. In: *Journal of Hydrology* 395.3-4 (2010), pp. 199–215.
- [9] Jordan G. Powers et al. “The Weather Research and Forecasting Model: Overview, System Efforts, and Future Directions”. In: *Bulletin of the American Meteorological Society* 98.8 (2017), pp. 1717–1737. DOI: [10.1175/BAMS-D-15-00308.1](https://doi.org/10.1175/BAMS-D-15-00308.1). eprint: <https://doi.org/10.1175/BAMS-D-15-00308.1>. URL: <https://doi.org/10.1175/BAMS-D-15-00308.1>.
- [10] Clarence W Richardson. “Stochastic simulation of daily precipitation, temperature, and solar radiation”. In: *Water resources research* 17.1 (1981), pp. 182–190.
- [11] DA Sachindra et al. “Statistical downscaling of general circulation model outputs to precipitation—part 1: calibration and validation”. In: *International Journal of Climatology* 34.11 (2014), pp. 3264–3281.
- [12] William C Skamarock et al. *A description of the advanced research WRF version 4*. Tech. rep. National Center For Atmospheric Research Boulder Co Mesoscale and Microscale ..., 2019.
- [13] Marek Smid and Ana Cristina Costa. “Climate projections and downscaling techniques: a discussion for impact studies in urban systems”. In: *International Journal of Urban Sciences* 22.3 (2018), pp. 277–307. DOI: [10.1080/12265934.2017.1409132](https://doi.org/10.1080/12265934.2017.1409132). eprint: <https://doi.org/10.1080/12265934.2017.1409132>. URL: <https://doi.org/10.1080/12265934.2017.1409132>.
- [14] T Sonkaew et al. “Finding the Optimum Microphysics and Convective Parameterization Schemes for the WRF Model for LPRU, Thailand”. In: ().

- [15] Robert L Wilby and Thomas ML Wigley. “Downscaling general circulation model output: a review of methods and limitations”. In: *Progress in physical geography* 21.4 (1997), pp. 530–548.