

Versi 1.0



# PANDUAN MENGGUNAKAN GrADS UNTUK PEMULA

*Membahas tentang basis dasar program GrADS ,  
instalasi GrADS, Tutorial, perancangan script  
program dan disertai dengan beberapa contoh  
script untuk aplikasi meteorologi dan klimatologi*



Pusat Klimatologi dan Kualitas Udara  
Badan Meteorologi dan Geofisika



# | K a t a Pengantar |

Puji syukur penulis panjatkan ke hadirat Allah SWT, Tuhan Yang Maha Esa yang telah memberikan taufik dan hidayah-Nya sehingga penulis dapat menyelesaikan buku ini yang berjudul “**PANDUAN MENGGUNAKAN GrADS UNTUK PEMULA**”.

Program GrADS adalah suatu program yang sangat populer yang digunakan untuk tampilan dan pengolahan data-data meteorologi, hal ini dapat dilihat bahwa lebih dari 50% jurnal-jurnal ilmiah meteorologi dan klimatologi menampilkan gambar yang diolah dengan menggunakan program GrADS ini.

Penulisan buku ini didasarkan atas kurangnya bahan referensi untuk mempelajari GrADS. Buku ini diharapkan dapat membantu pengguna dalam mengenal program GrADS dan dapat mengaplikasikannya dalam tugas sehari-hari. Pengenalan GrADS ini dibuat secara bertahap mulai dari instalasi, tutorial, pengenalan dan pembuatan script program serta dilengkapi dengan beberapa contoh script yang biasa dipergunakan di lingkungan meteorologi dan klimatologi.

Buku ini cocok untuk pemula dan menengah yang ingin mempelajari GrADS baik di lingkungan pegawai Badan Meteorologi dan Geofisika ataupun pihak-pihak lain yang berhubungan dengan lingkungan sains kebumihan seperti kelautan, pertanian dan lain-lain.

Dalam kesempatan ini penulis mengucapkan banyak terima kasih kepada Kepala Badan Meteorologi dan Geofisika (KBMG) Ibu Sriworo B. Harijono, MSc yang memberikan kesempatan kepada penulis untuk lebih mendalami GrADS, juga kepada Bapak Prof. DR. Mezak A. Ratag atas segala saran dan bimbingannya, Kapus Sisdatin Klimatologi dan KU, Kepala Bidang Analisa Klimatologi dan KU, Kasubid Analisa Kualitas Udara, Dr. Edvin Aldrian sebagai motivator, DR. Dodo Gunawan yang telah memberikan masukan dan seluruh pihak yang tidak dapat penulis sebutkan satu persatu. Serta terima kasih atas dukungan keluarga (Ratna Komala, Hafizh dan Faqih) semoga Allah selalu memberkahi kita semua. Amin.

Penulis menyadari bahwa buku ini jauh dari sempurna oleh karena itu penulis memohon kritik dan saran yang membangun untuk menyempurnakan buku ini di masa mendatang.

Jakarta, Agustus 2008  
Penulis,

Erwin Makmur, MSi

## **Kata Sambutan**

**Kepala Pusat SISDATIN Klimatologi dan Kualitas Udara**

# | Daftar Isi |

	Halaman
BAB I      PENDAHULUAN .....	4
BAB II     MENGGUNAKAN GrADS UNTUK PERTAMA KALI .....	5
BAB III    KONSEP OPERASI DASAR GrADS .....	12
BAB IV    TUTORIAL GrADS .....	13
BAB V     PENGGUNAAN SCRIPT PADA GrADS .....	49
BAB VI    PENUTUP .....	69
DAFTAR BACAAN .....	70
ALAMAT WEBSITE GrADS .....	70
LAMPIRAN – LAMPIRAN	
LAMPIRAN 1 GrADS Reference Card Version 1.7 .....	71
LAMPIRAN 2 GrADS Scripting Language Reference Card Version 1.7 .....	74

## BAB I

### PENDAHULUAN

The Grid Analysis and Display System (GrADS) merupakan *software* interaktif yang digunakan untuk memanipulasi dan visualisasi data sains kebumihan secara mudah. Grads dapat diperoleh dari internet secara bebas di <http://grads.iges.org/grads/downloads.html>.

Di dalam situs internet tersebut dapat dipilih Grads versi Windows yang tak perlu menggunakan *X server*. Xserver adalah *software* yang dapat meng-emulasi-kan tampilan monitor berbasis grafis.

GrADS merupakan *software* yang direkomendasikan oleh World Meteorological Organization (WMO) untuk menggambarkan parameter-parameter meteorologi dalam bentuk spasial dan jika kita memperhatikan jurnal-jurnal meteorologi internasional, sebagian besar gambar yang ditampilkan diolah menggunakan *software* GrADS.

Format data yang bisa digunakan dalam Grads adalah biner biasa, netCDF, dan HDF-SDS (*Hierarchical Data Format – Scientific Data Format*).

Grads dapat menggunakan data dengan 4 dimensi: garis bujur, garis lintang, ketinggian (level), dan waktu.

Data dapat ditampilkan menggunakan bermacam teknik grafis seperti grafik garis, grafik batang, kontur biasa, kontur berwarna, vektor angin, ataupun garis alur (*streamlines*).

Penggunaan tipe grafik yang digunakan tergantung pada jenis variabel yang ingin ditampilkan. Untuk curah hujan dapat digunakan kontur berwarna dan untuk angin digunakan vektor angin.

## BAB II MENGUNAKAN GrADS UNTUK PERTAMA KALI

### Download *Software* GrADS dari Internet

Apabila kita belum mempunyai *software* GrADS, maka kita dapat men-*download* secara gratis di website GrADS di alamat internet:

<http://www.iges.org/grads/downloads.html>. Jika kita masuk ke alamat tersebut maka akan diberikan beberapa pilihan *software* GrADS. Pilih *software* GrADS versi MS Windows, klik pada tulisan win32e (lihat gambar 2.1).

**Earlier Releases of GrADS**  
 The GrADS distributions for version 1.8 and 1.9 contain pre-compiled binary executables, the source code, documentation, and the supplementary data sets that are required to run GrADS (fonts and map files). Two MS Windows builds of version 1.8 are available: xwin32 requires an X-window server in order to display graphics, and win32e uses native windows. Click on the "Getting Started" links for more information about the win32 builds.

Hardware / Operating System	GrADS 1.8s11	GrADS 1.9b4
	<a href="#">source</a>	<a href="#">source</a>
Linux	<a href="#">linux</a>	<a href="#">linuxRH7.1</a> <a href="#">linuxRH9</a> <a href="#">linuxRHE3</a>
SUN	<a href="#">sol55</a>	<a href="#">sun59</a>
Macintosh OSX	<a href="#">darwin</a>	<a href="#">darwin</a>
SGI / IRIX	<a href="#">irix6</a>	<a href="#">irix6</a>
DEC	<a href="#">alpha</a>	<a href="#">alpha</a>
IBM / AIX	<a href="#">aix</a>	
MS Windows	<a href="#">xwin32</a> (Requires an X server; <a href="#">Getting xwin32 started</a> ) <a href="#">win32e</a> (Uses native windows; <a href="#">Getting win32e started</a> )	

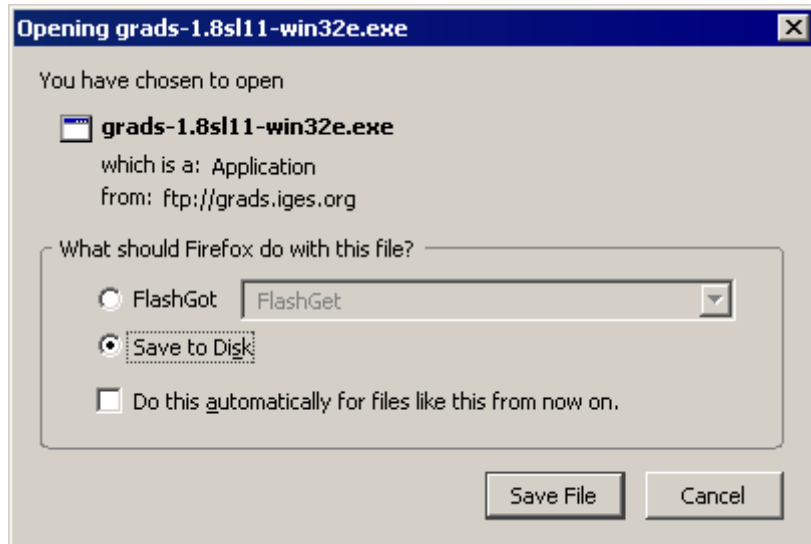
**Klik di sini** →

what Executable Files are in a Release?

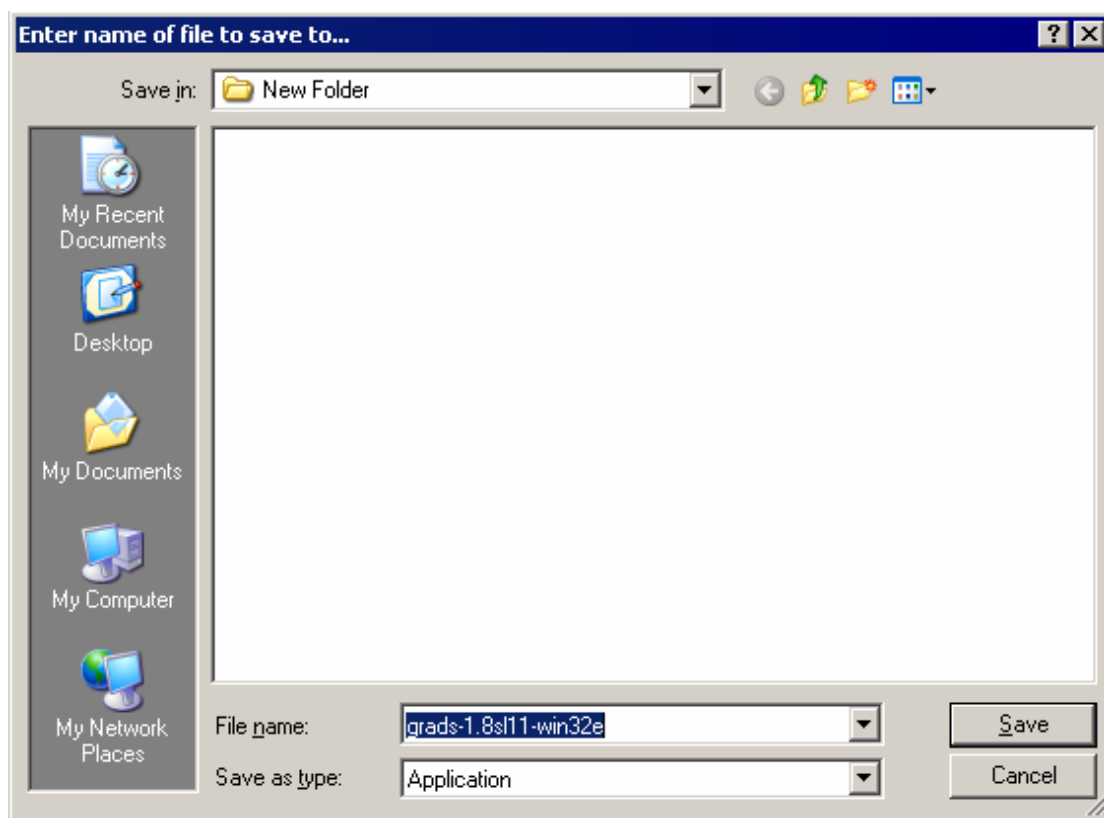
Gambar 2.1

Jika diklik maka akan muncul pilihan seperti pada gambar 2.2. Cek pada kotak *Save to Disk* dan kemudian klik tombol *Save File*. Maka akan ditanyakan di mana file tersebut akan disimpan (lihat gambar 2.3). Maka kita simpan pada lokasi yang kita inginkan dengan mengetiknya di kolom *File name* dan klik *Save*. Maka proses *download* akan berlangsung dan kita tunggu sampai proses *download* tersebut sempurna. Dikarenakan *file* tersebut cukup besar maka waktu *download* tersebut akan bervariasi tergantung dari kecepatan koneksi internetnya.

Dari website GrADS tersebut juga dapat di-*download* beberapa *software* lain seperti gv32.exe dan file gv32.hlp yang merupakan software untuk menampilkan dan meng-*edit* tampilan GrADS. Sebaiknya file gv tersebut juga di-*download*.



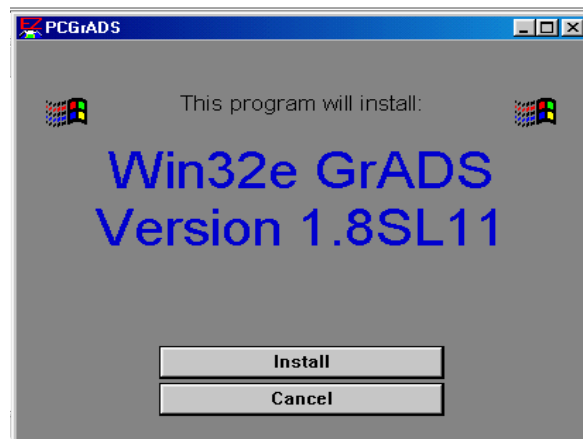
Gambar 2.2



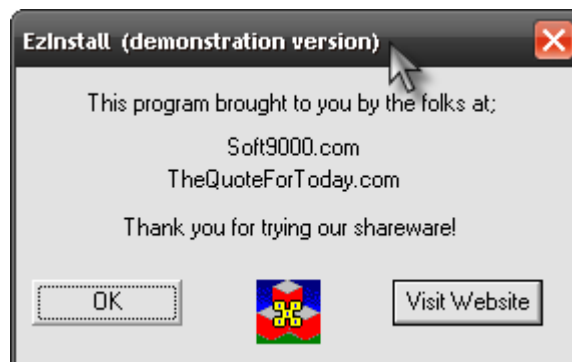
Gambar 2.3

## Instalasi GrADS

Setelah kita mendapatkan *software* grads dari internet, versi paling baru adalah *grads-1.8sl11-win32e.exe* untuk Windows 98. Kita jalankan saja file .exe tersebut dari windows dan kita akan mendapatkan tampilan seperti berikut. Ikuti perintah yang diberikan, misalnya dalam tampilan di bawah ini kita klik ***install*** jika kita memang ingin meng-*install*-nya. Dalam meng-*install* GrADS kita tinggal mengikuti perintah-perintahnya dan tidak perlu mengetikkan serial number dikarenakan *software* GrADS adalah *software* bebas.



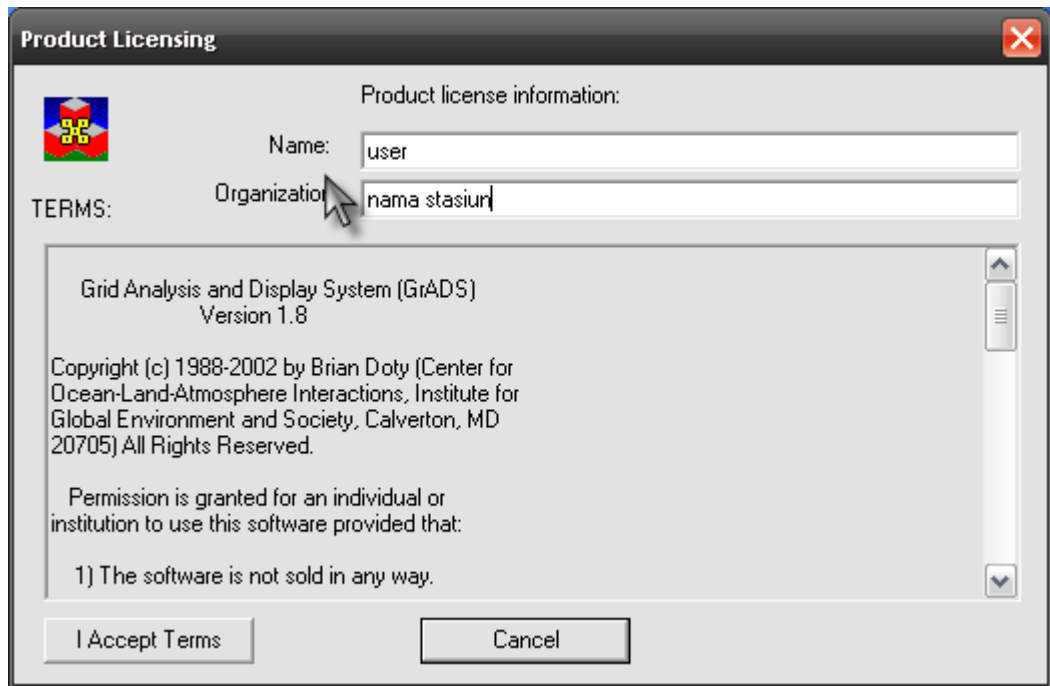
Gambar 2.4



Gambar 2.5

Klik tombol **OK**

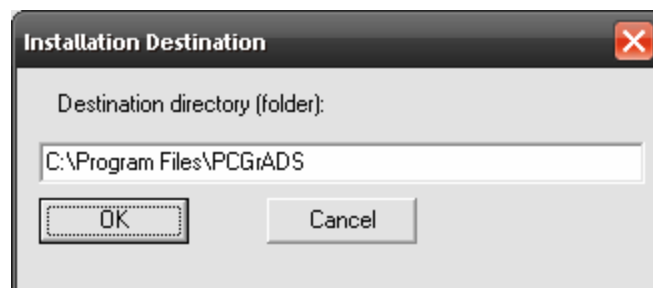




Gambar 2.6

Pada kolom Name ketikkan nama *user* (misal “user”) dan pada kolom *Organization* ketikkan nama stasiun Anda (catatan : kedua kolom tidak wajib diisi)

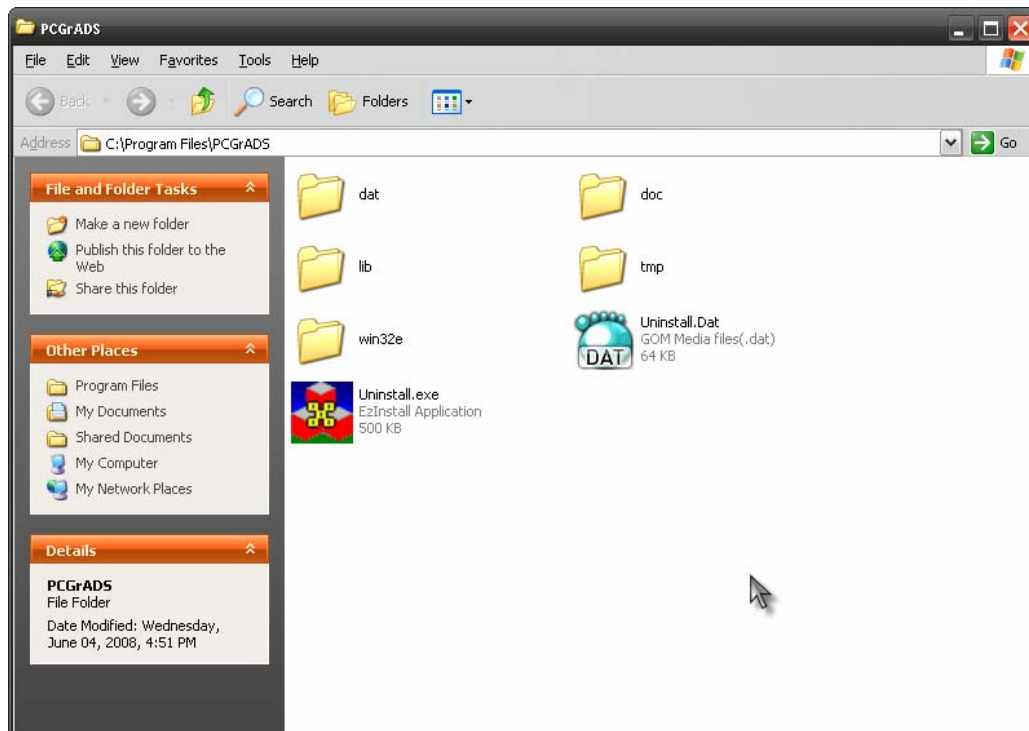
Setelah itu klik tombol **I Accept Terms**



Gambar 2.7

Klik tombol **OK**

Kemudian proses instalasi segera diproses, tunggu sampai muncul jendela berikut :

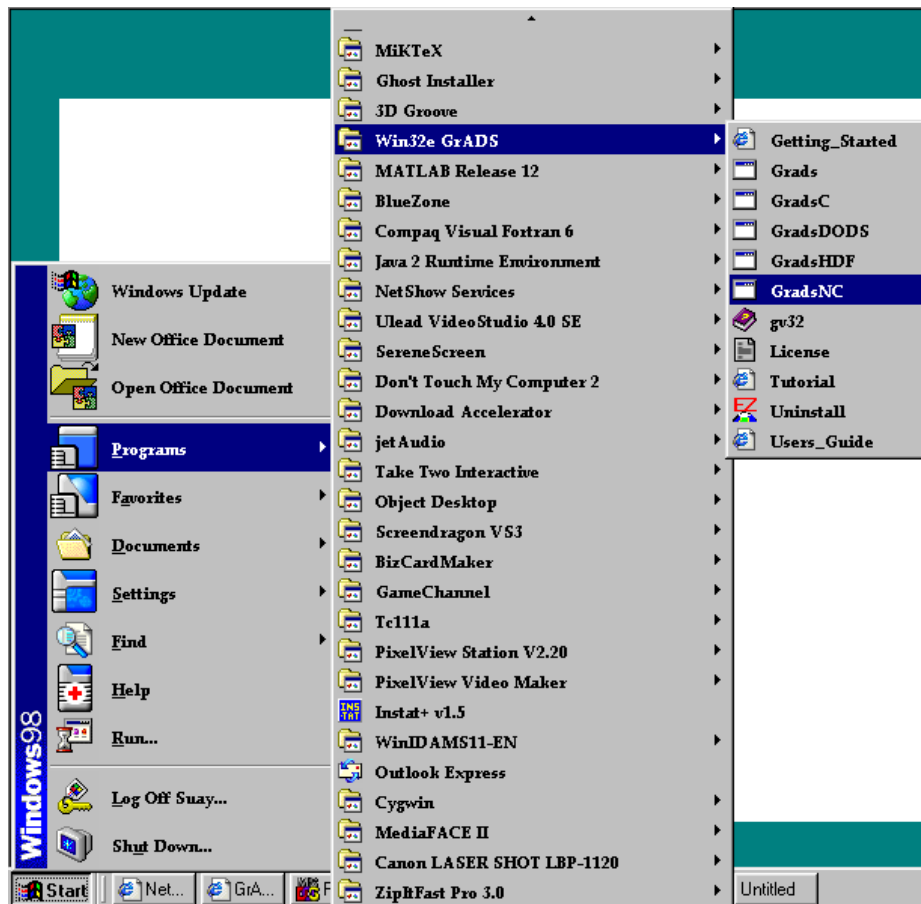


Gambar 2.8

Jika jendela tersebut telah muncul artinya *software* GrADS telah ter-*install* dengan sempurna dan kita siap untuk menjalankan GrADS.

Selesai install, kita jalankan grads dengan mencari di menu start program windows dan kita pilih perintah Grads (seperti terlihat pada gambar di bawah ini), karena data yang akan ditampilkan adalah data biner.

Kalau kita menjalankan perintah GradsNC, kita bisa membuka file netCDF dan file biner biasa. Sedangkan kalau kita jalankan perintah Grads atau GradsC, kita hanya bisa membuka file biner biasa dan tidak dapat membuka file netCDF. Kita akan diberi pilihan mode *landscape* (mendatar) atau *portrait* (tegak). Langsung tekan enter jika ingin gambar mendatar, ketik no dan tekan enter kalau ingin mode tegak. Gambar di bawah adalah grads dalam mode mendatar.



Gambar 2.9

Setelah menjalankan GrADS maka akan terlihat dua buah jendela yaitu jendela untuk mengetikkan perintah dan jendela display, dimana perintah yang kita ketikkan di jendela perintah akan dilihat hasilnya pada jendela display

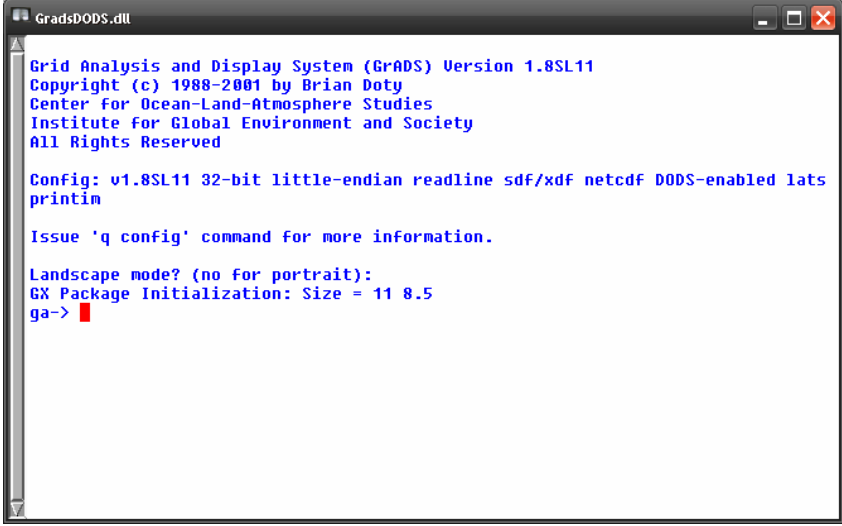
Di dalam tampilan GrADS kita akan melihat prompt yang GrADS,

ga- >

di dalam prompt tersebut kita dapat mengetikkan perintah-perintah GrADS

Untuk keluar dari jendela GrADS ketikkan :

ga- >quit



```
GradsDODS.dtl
Grid Analysis and Display System (GrADS) Version 1.8SL11
Copyright (c) 1988-2001 by Brian Doty
Center for Ocean-Land-Atmosphere Studies
Institute for Global Environment and Society
All Rights Reserved

Config: v1.8SL11 32-bit little-endian readline sdf/xdf netcdf DODS-enabled lats
printin

Issue 'q config' command for more information.

Landscape mode? (no for portrait):
GX Package Initialization: Size = 11 8.5
ga-> █
```

Gambar 2.10

### BAB III

## KONSEP OPERASI DASAR GrADS

Terdapat 3 (tiga) perintah dasar GrADS, yaitu :

- **open** adalah perintah untuk membuka file grid atau data stasiun.
- **d** untuk menggambarkan ekspresi GrADS.
- **set** adalah perintah untuk memanipulasi “apa”, “dimana” dan “bagaimana” data digambarkan.

Selanjutnya ketiga perintah dasar tersebut akan dibahas lebih detail dalam bab tutorial berikut ini.

Ekspresi GrADS atau “**apa**” yang Anda ingin lihat dapat dijadikan sesuatu yang sederhana dari variabel pada file data yang telah dibuka misalnya untuk menampilkan suhu cukup mengetikkan ‘**d tmp**’ atau juga dapat memasukkan operasi aritmetika seperti ‘**d tmp-273.15**’ yaitu untuk menampilkan variabel suhu dalam satuan derajat celcius atau suatu perintah yang telah disiapkan oleh GrADS seperti perintah **mag** misalnya ‘**d mag(ugrd,vgrd)**’ yaitu untuk menampilkan magnitudo dari variabel angin, dimana **mag(ugrd,vgrd)** merupakan formula dari  $\sqrt{ugrd*ugrd+vgrd*vgrd}$ . “**Di mana**” dari tampilan data disebut “**dimensi lingkungan**” dan mendefinisikan bagian mana, potongan atau irisan dari ruang geofisikal 4D (bujur, lintang, ketinggian dan waktu) yang diinginkan. Dimensi lingkungan ini diatur dengan perintah **set** dan dikontrol oleh salah satunya adalah koordinat grid (x, y, z, t atau indeks) atau koordinat dunia (bujur, lintang, ketinggian dan waktu).

“**Apa**” dan “**Bagaimana**” dari tampilan dikontrol oleh perintah **set** dan terdapat didalamnya baik metode grafik (misalnya kontur, streamline) dan data (misalnya display untuk suatu file)

Grafik GrADS dapat ditulis kembali menjadi suatu file (dengan perintah **enable print filename dan print**) dan kemudian dikonversi menjadi file PostScript untuk pencetakan atau dikonversi menjadi format lain misalnya .gif.

Sebagai tambahan lainnya, GrADS termasuk primitif grafik (misalnya garis dan lingkaran) dan pelabelan dapat dilakukan dengan perintah **draw**.

Perintah **q** atau **query** digunakan untuk mendapatkan informasi dari GrADS tentang file yang dibuka dan data statistiknya.

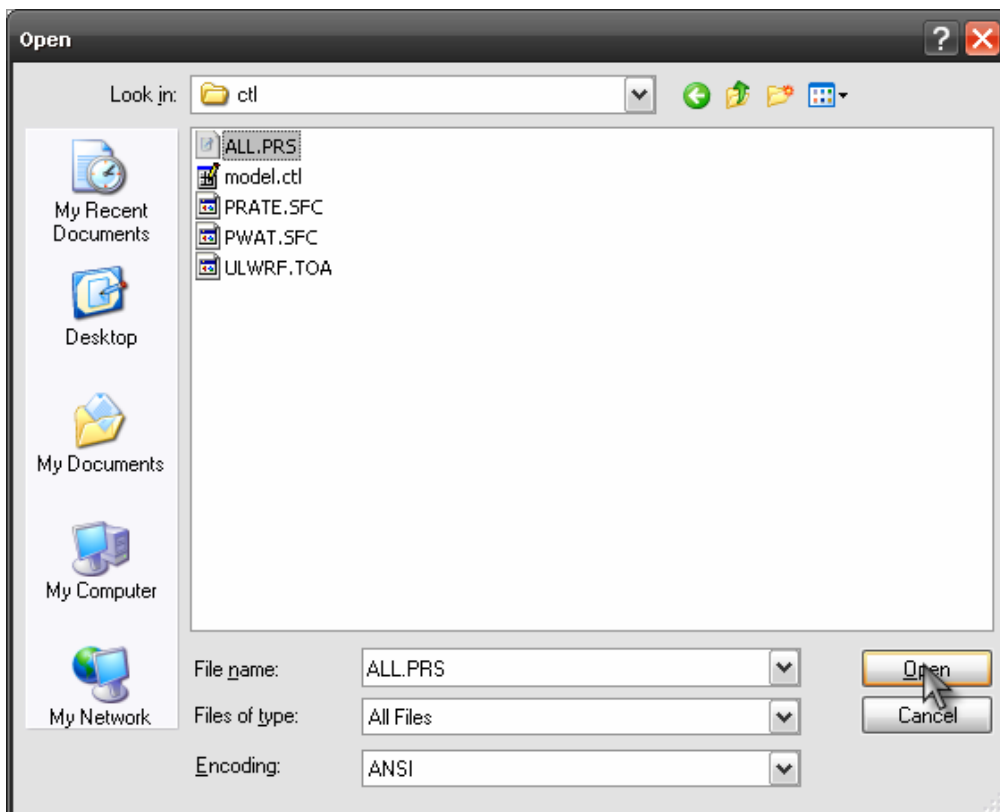
## BAB IV TUTORIAL GrADS

Sebelum memulai tutorial pastikan bahwa *software* GrADS telah ter-*install* di komputer Anda dan sebagai untuk data latihan terdapat beberapa file berikut yang telah dicopykan di komputer Anda.

<b>all.prs</b>	yang merupakan file deskriptor	(2 kb)
<b>all.prs</b>	yang merupakan file data	(41 mb)
<b>all.prs</b>	yang merupakan file index	(55 kb)

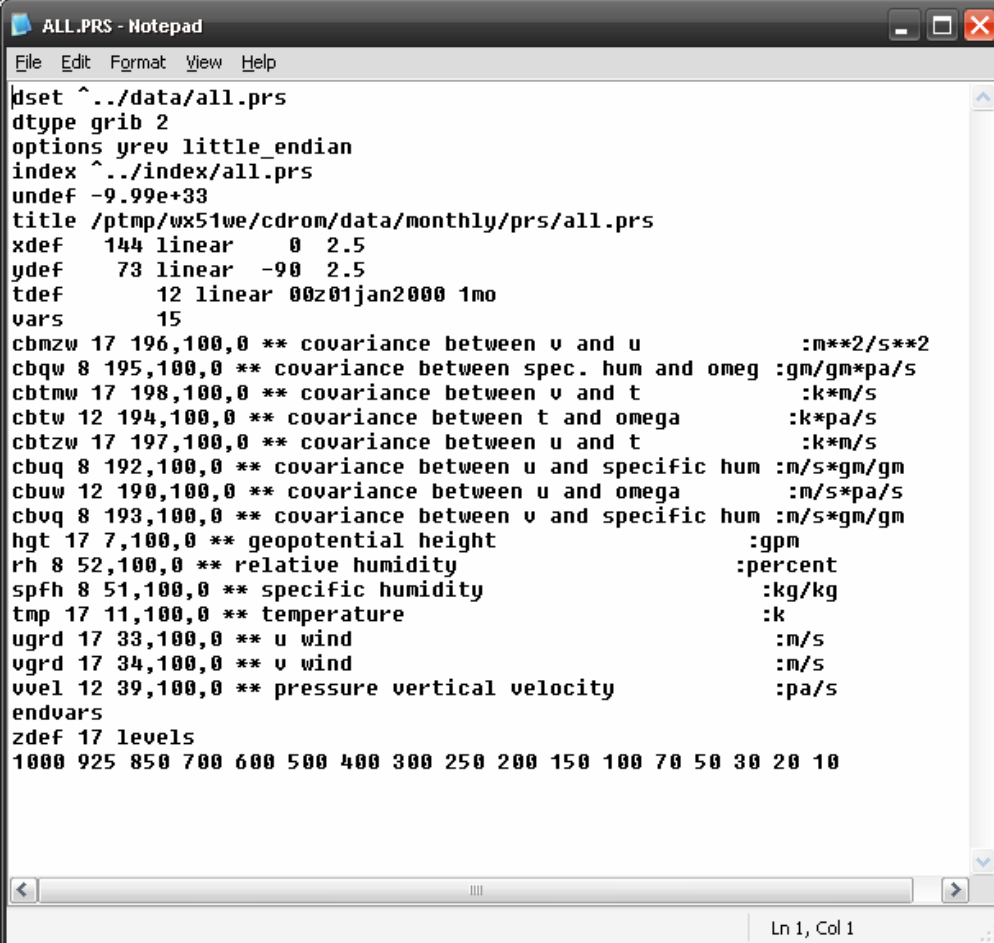
Ketiga file tersebut mempunyai nama yang sama tetapi mempunyai format dan fungsi yang berbeda. Untuk menjalankan tutorial ini ketiga file tersebut harus lengkap disalin pada direktorinya masing-masing.

Untuk melihat isi dari file deskriptor (*all.prs*) dapat kita buka dengan mempergunakan program notepad atau program pengolah kata lainnya.



Gambar 4.1

Klik [Open](#), maka akan terlihat isi file yaitu sebagai berikut :



```

ALL.PRS - Notepad
File Edit Format View Help
dset ^../data/all.prs
dtype grib 2
options yrev little_endian
index ^../index/all.prs
undef -9.99e+33
title /ptmp/wx51we/cdrom/data/monthly/prs/all.prs
xdef 144 linear 0 2.5
ydef 73 linear -90 2.5
tdef 12 linear 00z01jan2000 1mo
vars 15
cbmw 17 196,100,0 ** covariance between v and u :m**2/s**2
cbqw 8 195,100,0 ** covariance between spec. hum and omeg :gm/gm*pa/s
cbtmw 17 198,100,0 ** covariance between v and t :k*m/s
cbtw 12 194,100,0 ** covariance between t and omega :k*pa/s
cbtz 17 197,100,0 ** covariance between u and t :k*m/s
cbuq 8 192,100,0 ** covariance between u and specific hum :m/s*gm/gm
cbuw 12 190,100,0 ** covariance between u and omega :m/s*pa/s
cbvq 8 193,100,0 ** covariance between v and specific hum :m/s*gm/gm
hgt 17 7,100,0 ** geopotential height :gpm
rh 8 52,100,0 ** relative humidity :percent
spfh 8 51,100,0 ** specific humidity :kg/kg
tmp 17 11,100,0 ** temperature :k
ugrd 17 33,100,0 ** u wind :m/s
vgrd 17 34,100,0 ** v wind :m/s
vvel 12 39,100,0 ** pressure vertical velocity :pa/s
endvars
zdef 17 levels
1000 925 850 700 600 500 400 300 250 200 150 100 70 50 30 20 10
Ln 1, Col 1

```

Gambar 4.2

Berikut ini adalah beberapa penjelasan tentang fungsi dari fungsi-fungsi tersebut :

```

dset ^../data/all.prs      % menunjukkan lokasi dari file data
dtype grib 2              % menunjukkan data merupakan data tipe GRIB 2
options yrev little_endian
index ^../index/all.prs   % menunjukkan lokasi dari file index
undef -9.99e+33           % jika data kosong ditunjukkan oleh nilai -9.99e+33
title /ptmp/wx51we/cdrom/data/monthly/prs/all.prs
xdef 144 linear 0 2.5     % level untuk x (arah mendatar sumbu x)
ydef 73 linear -90 2.5   % level untuk y (arah mendatar sumbu y)
tdef 12 linear 00z01jan2000 1mo % level waktu (12 bulan)
vars 15                  % jumlah variabel meteorologi
cbmw 17 196,100,0 ** covariance between v and u :m**2/s**2
cbqw 8 195,100,0 ** covariance between spec. hum and omeg :gm/gm*pa/s
cbtmw 17 198,100,0 ** covariance between v and t :k*m/s

```

```
cbtw 12 194,100,0 ** covariance between t and omega      :k*pa/s
cbtz 17 197,100,0 ** covariance between u and t          :k*m/s
cbuq 8 192,100,0 ** covariance between u and specific hum :m/s*gm/gm
cbuw 12 190,100,0 ** covariance between u and omega      :m/s*pa/s
cbvq 8 193,100,0 ** covariance between v and specific hum :m/s*gm/gm
hgt 17 7,100,0 ** geopotential height                   :gpm
rh 8 52,100,0 ** relatif humidity                       :percent
spfh 8 51,100,0 ** specific humidity                    :kg/kg
tmp 17 11,100,0 ** temperatur                           :k
ugrd 17 33,100,0 ** u wind                              :m/s
vgrd 17 34,100,0 ** v wind                              :m/s
vvel 12 39,100,0 ** pressure vertical velocity          :pa/s
endvars
zdef 17 levels          % level untuk arah vertikal (sumbu z)
1000 925 850 700 600 500 400 300 250 200 150 100 70 50 30 20 10
```

Kita dapat mengedit file deskriptor tersebut, biasanya untuk merubah lokasi file data dan file index jika tidak satu directory dengan file deskriptornya. Jangan lupa disimpan kembali dengan menekan tombol Ctrl-S secara bersamaan.

Jika telah disimpan maka kita telah siap untuk menjalankan tutorial berikut.

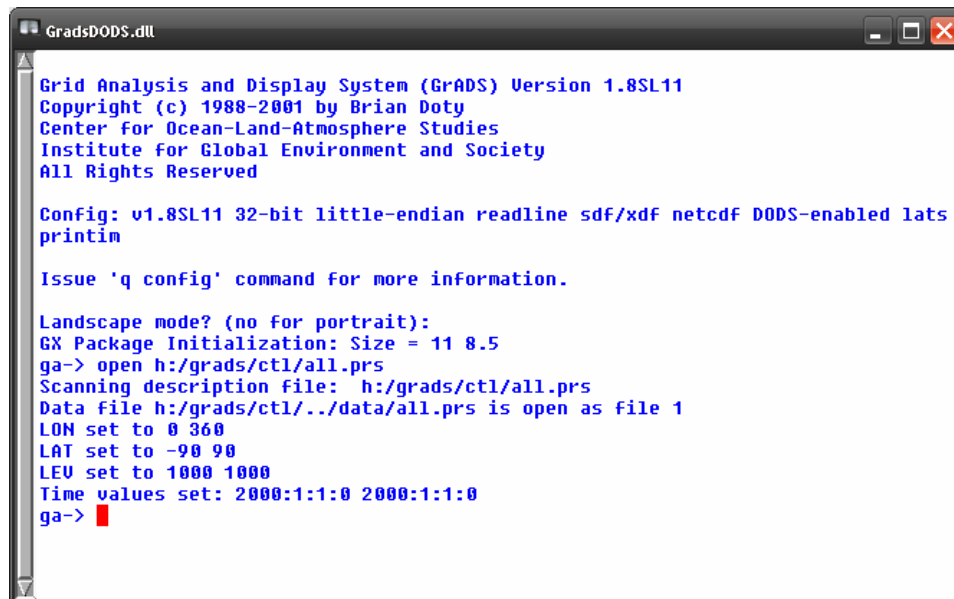
a. Membuka file deskriptor

Ketik open diikuti lokasi file dan nama file deskriptor

ga- >open h:/grads/ctl/all.prs

maka akan muncul penjelasan dari file yang telah kita buka





```
GradsDODS.dll
Grid Analysis and Display System (GrADS) Version 1.8SL11
Copyright (c) 1988-2001 by Brian Doty
Center for Ocean-Land-Atmosphere Studies
Institute for Global Environment and Society
All Rights Reserved

Config: v1.8SL11 32-bit little-endian readline sdf/xdf netcdf DODS-enabled lats
printim

Issue 'q config' command for more information.

Landscape mode? (no for portrait):
GX Package Initialization: Size = 11 8.5
ga-> open h:/grads/ctl/all.prs
Scanning description file: h:/grads/ctl/all.prs
Data file h:/grads/ctl/./data/all.prs is open as file 1
LON set to 0 360
LAT set to -90 90
LEU set to 1000 1000
Time values set: 2000:1:1:0 2000:1:1:0
ga-> █
```

Gambar 4.2

setelah sukses membuka file, maka kita dapat melihat semua parameter meteorologi yang dikandung oleh file tersebut dengan mengetikkan :

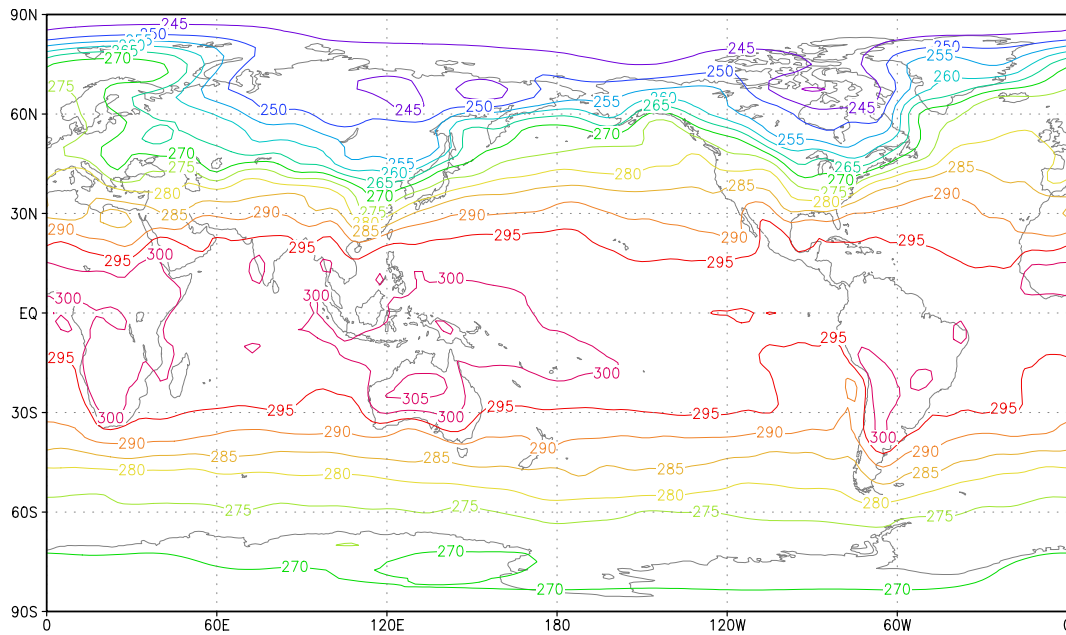
ga- >q file (enter)      atau      ga- >query file (enter)

dimana pada jendela GrADS kita akan melihat 15 variabel meteorologi, misalnya terlihat variabel tmp yang merupakan variabel suhu dalam satuan derajat kelvin, jika ingin menampilkan variabel suhu tersebut ketikkan :

ga- >d tmp (enter)

perintah **d** merupakan singkatan dari **display**. Sebagai nilai awal (default) maka GrADS akan menampilkan plot dalam sumbu X, Y pada waktu pertama dan pada level terbawah dari dataset, maka akan terlihat tampilan pada jendela display seperti pada gambar 4.3.

Jika kita belum merubah setting awal maka GrADS akan memplot tampilan pada peta dunia.



GrADS: COLA/IGES

2008-08-28-17:51

Gambar 4.3

Sekarang jika kita ingin merubah dimensi dari gambar untuk mendapatkan nilai data pada satu grid saja maka dimensi yang ditentukan harus tidak bervariasi, misalnya

<a href="#">clear</a> atau <a href="#">c</a>	% menghapus display sebelumnya (bersihkan layar)
<a href="#">set lon</a> 90	% set bujur ke 90°BT
<a href="#">set lat</a> 0	% set lintang ke 0° (ekuator)
<a href="#">set lev</a> 500	% set ketinggian ke 500 mb
<a href="#">set t</a> 1	% set waktu ke waktu pertama (bulan Januari 2000)
<a href="#">d tmp</a>	% menampilkan variabel suhu

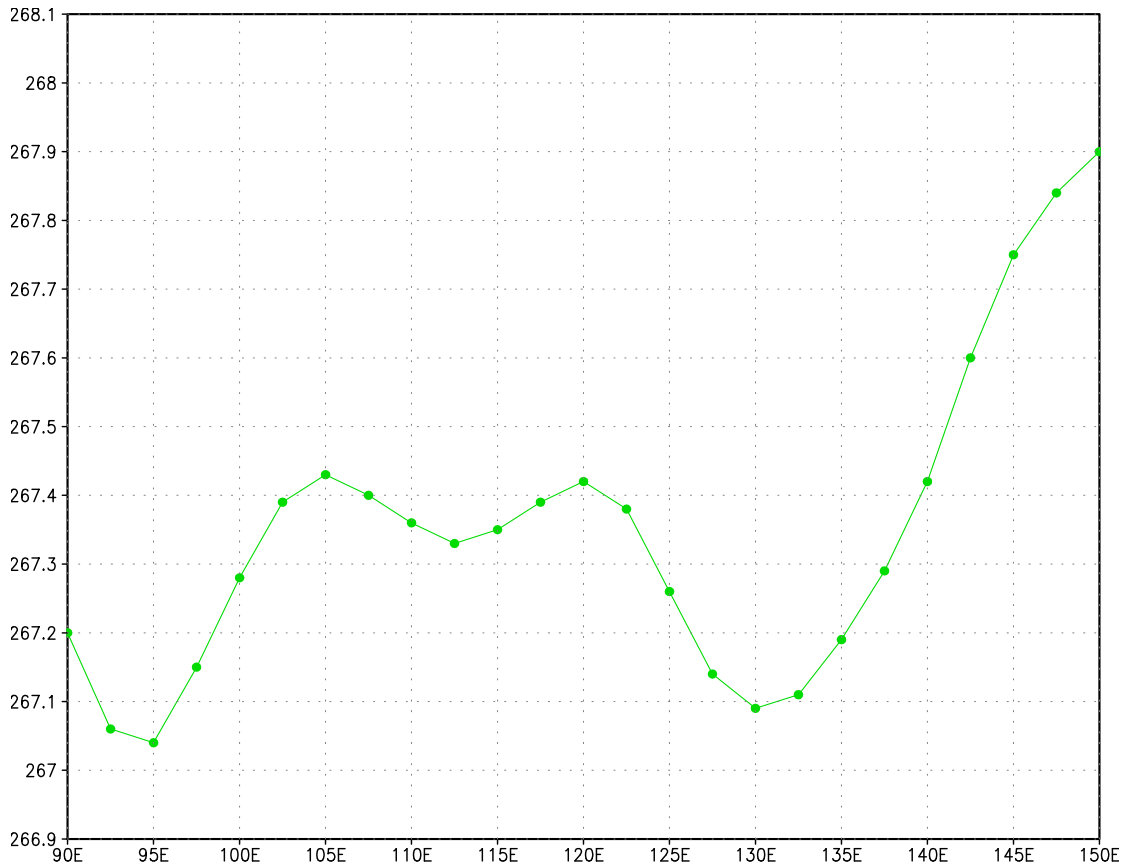
Pada urutan perintah di atas semua dimensi GrADS diset ke nilai tunggal. Jika diset ke nilai tunggal maka dimensi tersebut dinamakan “dimension fixed” dan akan mendapatkan variabel yang juga tunggal.

Jika kita mengetikkan urutan perintah tersebut akan didapatkan harga suhu pada bujur 90°BT, lintang 0°, ketinggian 500 mb pada bulan Januari 2000 adalah sebesar **267.2°K**.

Sekarang jika mengetikkan :

```
set lon 90 150      % X bervariasi dari 90 BT sd 150 BT  
d tmp
```

Kita sekarang mengeset X bervariasi tetapi dimensi lintang, ketinggian dan waktu pada nilai tunggal (lintang  $0^\circ$ , ketinggian 500mb pada bulan Januari 2000) maka didapatkan grafik seperti pada gambar 4.4 berikut.



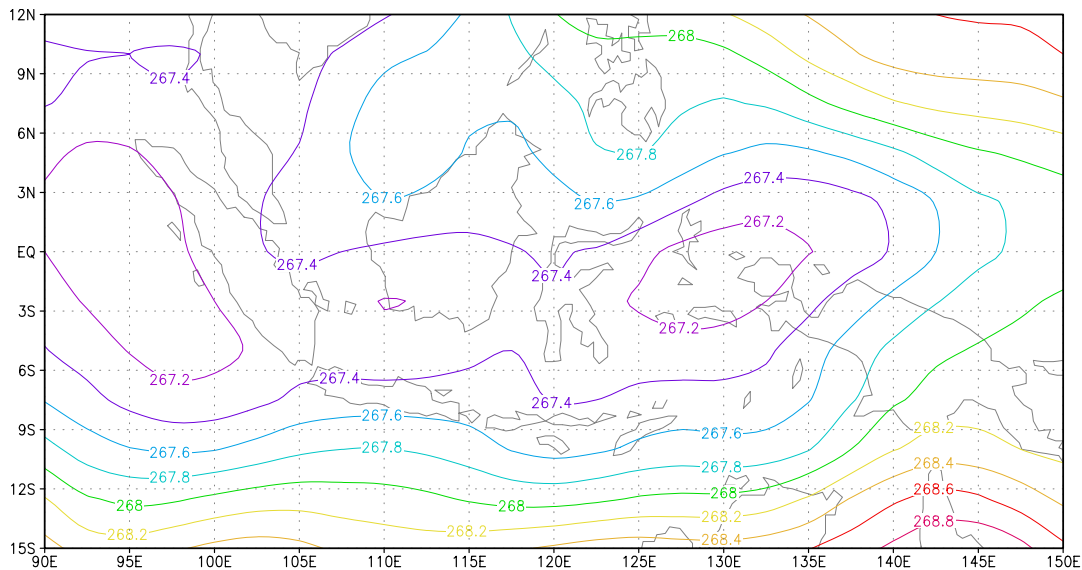
Gambar 4.4

Berikutnya ketikkan :

```
clear  
set lat -15 12      Set lintang dari 15°LS sd 12°LU  
d tmp
```

didapatkan grafik seperti pada gambar 4.5.

Didapatkan grafik suhu pada bujur  $90^\circ\text{BT}$ - $150^\circ\text{BT}$ ,  $15^\circ\text{LS}$ - $12^\circ\text{LU}$ , ketinggian 500 mb pada bulan Januari 2000



GrADS: COLA/IGES

2008-08-28-17:56

Gambar 4.5

Bagaimana jika kita set variabel waktu juga bervariasi misalnya dari bulan Januari 2000 sd Desember 2000, maka kita ketikkan :

```
c  
set t 1 12  
d tmp
```

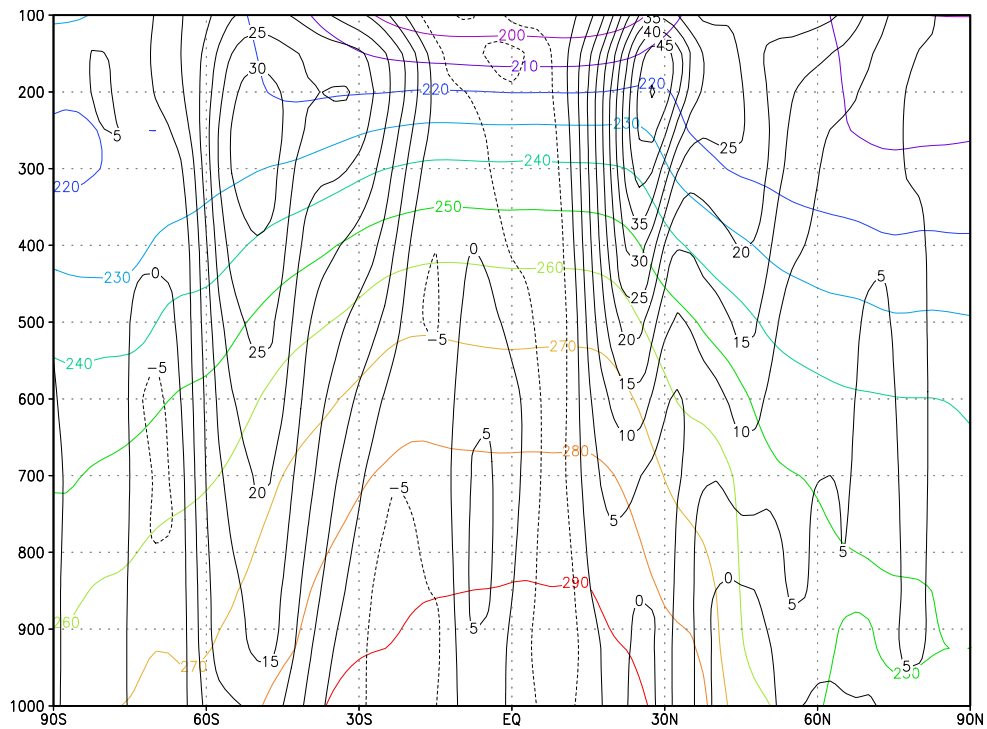
Maka akan didapatkan gambar animasi (12 grafik) yang masing-masing gambar mewakili setiap bulannya.

Berikutnya coba ketikkan perintah berikut :

```
clear  
set lon 90  
set lat -90 90  
set lev 1000 100  
set t 1  
d tmp  
d ugrd
```

Pada kasus ini kita mengeset dimensi Lintang (sumbu Y) dan ketinggian (sumbu Z) bervariasi, maka akan didapatkan cross section vertikal. Kita juga menampilkan 2 (dua) variabel secara bersamaan yaitu variabel suhu dan angin zonal. Untuk menampilkan variabel secara overlay kita cukup mengetikkan

perintah [display](#) diikuti nama variabel-variabel sebelum mengetik perintah [clear](#) lihat gambar 4.6.



GrADS: COLA/IGES

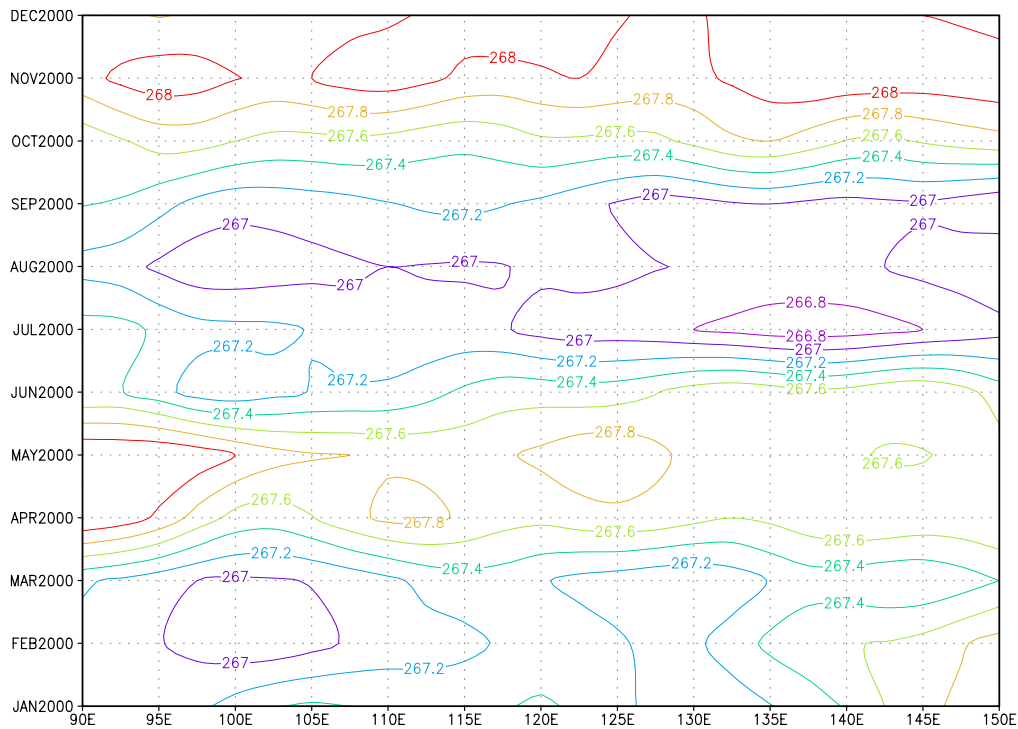
2008-08-28-19:45

Gambar 4.6

Contoh lainnya apa yang dinamakan dengan diagram **Hovmoller** yaitu X dan t (waktu) bervariasi

```
c  
set lon 90 150  
set lat 0  
set lev 500  
set t 1 12  
d tmp
```

Diagram ini untuk melihat penjalaran suatu variabel secara horizontal dengan perubahan waktu. (gambar 4.7)



GrADS: COLA/IGES

2008-08-28-19:45

Gambar 4.7

Dari penjelasan di atas diharapkan pembaca memahami bagaimana menampilkan beberapa parameter meteorologi dengan tampilan yang berbeda-beda disesuaikan dengan tujuan yang diharapkan. Selanjutnya akan dilanjutkan dengan operasi terhadap data. Selain mempunyai kemampuan untuk menampilkan parameter meteorologi, GrADS juga mempunyai kemampuan untuk mengolah suatu data baik manipulasi ataupun operasi matematis.

Pertama kita akan coba mengkonversi satuan suhu, dimana pada awalnya suhu yang tampilkan satuannya adalah dalam derajat Kelvin, sekarang kita ingin coba merubah satuan tersebut dalam derajat Fahrenheit. Set dimensi lingkungan sebagai berikut :

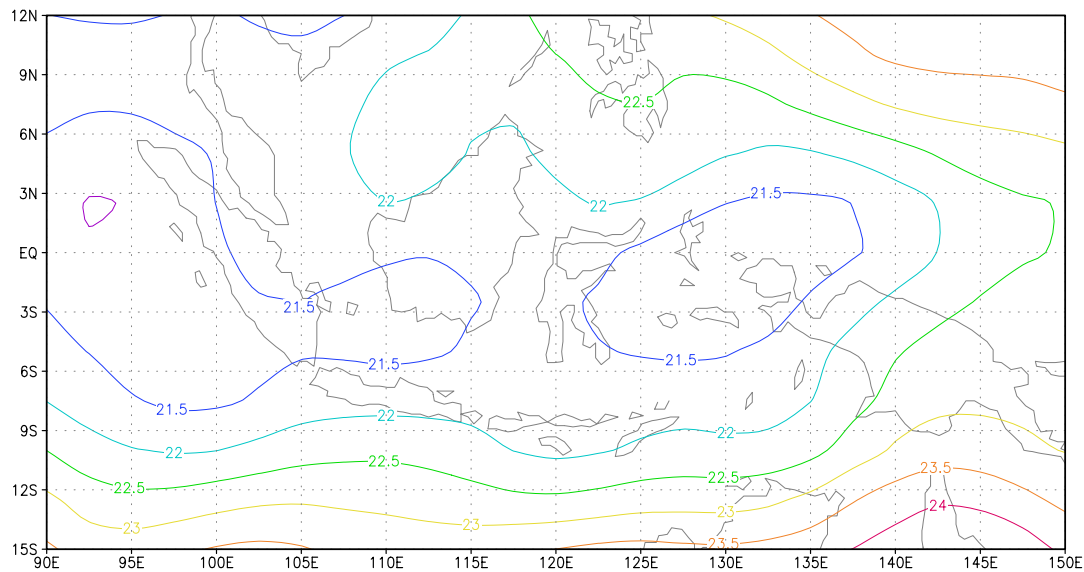
```
clear
set lon 90 150
set lat -15 12
set lev 500
set t 1
```

setelah itu kita masukkan formula yang mengkonversi satuan dari derajat Kelvin ke derajat Fahrenheit yaitu

`display (tmp-273.15)*9/5+32`

Berbagai ekspresi matematika dapat dipergunakan dalam operasi pada GrADS ini seperti operasi standar +, -, \*, dan /, serta juga dapat memasukkan suatu konstanta, variabel ataupun fungsi.

Jika kita jalankan maka akan dihasilkan grafik sebagai berikut :



GrADS: COLA/IGES

2008-08-28-19:45

Gambar 4.8

Selanjutnya untuk menampilkan kekuatan atau magnitudo dari angin dapat dipergunakan formula sebagai berikut :

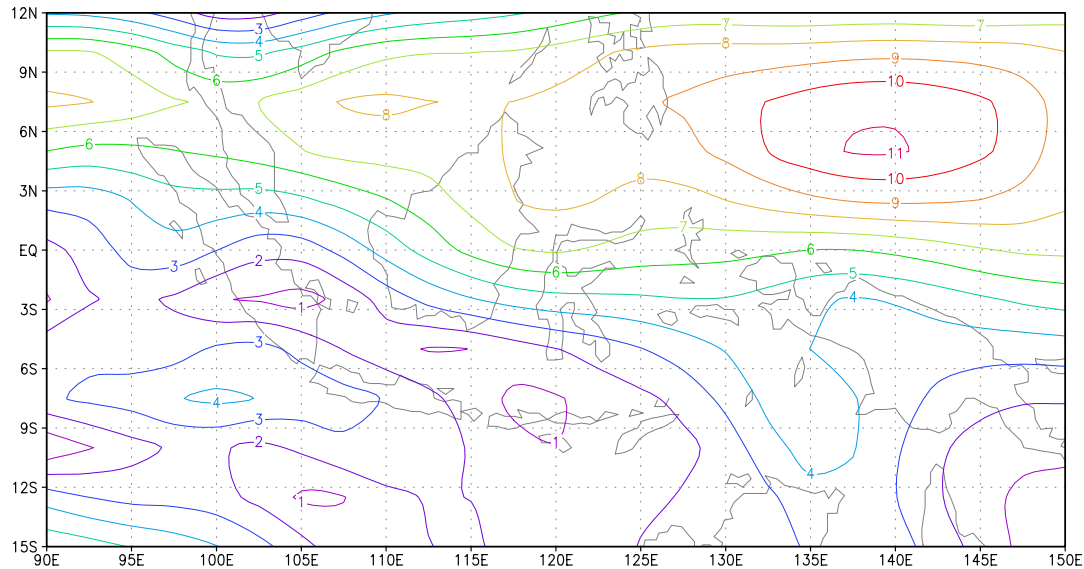
`clear`

`d sqrt(ugrd*ugrd+vgrd*vgrd)`

dimana ugrd adalah angin zonal (angin yang bergerak ke arah timur – barat), sedangkan vgrd adalah angin meridional (angin yang bergerak ke arah utara – selatan). Magnitudo dari angin merupakan besar resultan antara angin zonal dan meridional tersebut. Dalam perintah GrADS formula `sqrt(ugrd*ugrd+vgrd*vgrd)` dapat disederhanakan menjadi sebagai berikut :

`d mag(ugrd,vgrd)`

Maka akan didapatkan grafik seperti pada gambar 4.9.



GrADS: COLA/IGES

2008-08-28-19:45

Gambar 4.9

Dari grafik di atas terlihat daerah-daerah yang mempunyai kekuatan angin yang sama.

Selanjutnya kita akan mempergunakan suatu perintah GrADS yang berguna untuk membuat suatu perata-rataan terhadap suatu parameter meteorologi, misalkan kita ingin membuat rata-rata angin zonal di wilayah Indonesia pada bulan Januari, Februari, Maret (rata-rata 3 bulanan), maka ketikkan perintah sebagai berikut :

`clear`  
`d ave(ugrd,t=1,t=3)`

Secara umum syntax dari perintah ave() adalah sebagai berikut :

`ave(expr, dim1, dim2 <,tinc> <,-b>)`

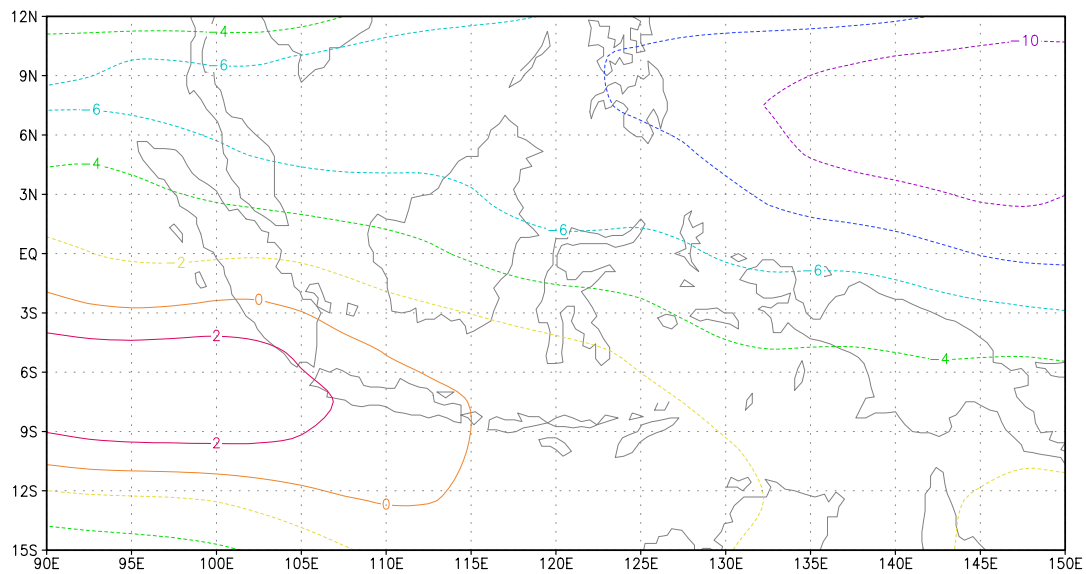
*expr* - ekspresi GrADS (parameter meteorologi yang ingin dibuat rata-ratanya)

*dim1* - Titik awal perata-rataan

*dim2* - Titik akhir perata-rataan



- `tinc` - pilihan tambahan untuk waktu perata-ratan (optional)
- `-b` - menggunakan batas eksak (optional)



GrADS: COLA/IGES

2008-08-28-19:45

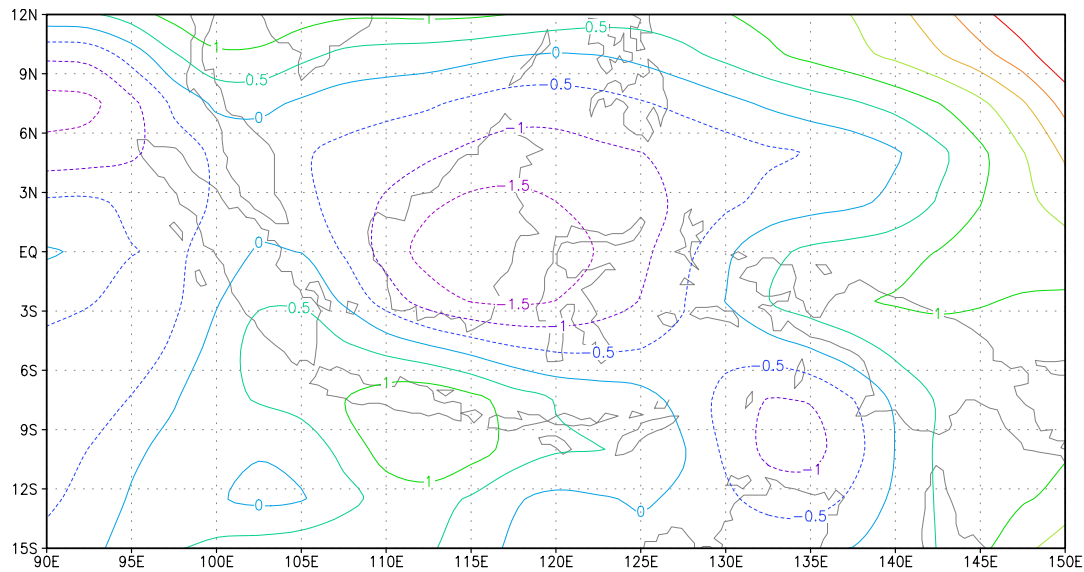
Gambar 4.10

kita juga dapat menghilangkan rata-rata dari field yang sedang diaktikan dengan cara

`clear`

`d ugrd - ave(ugrd,t=1,t=3)`

maka akan didapatkan grafik seperti terlihat pada gambar 4.11



GrADS: COLA/IGES

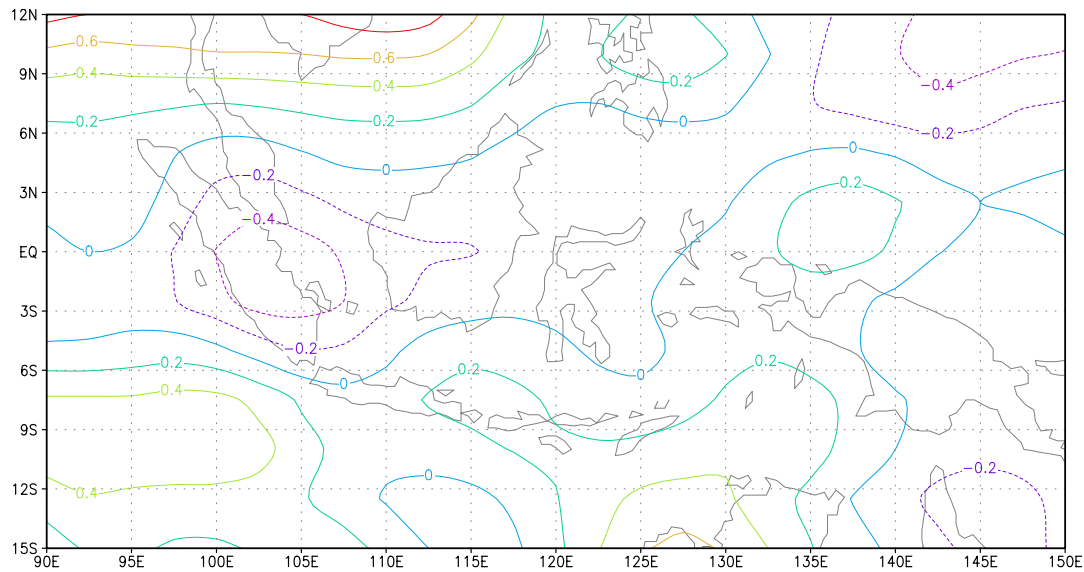
2008-08-28-19:45

Gambar 4.11

Kita dapat menampilkan perbedaan antara temperatur zonal pada bulan Februari dan bulan Januari dengan mengetikkan :

```
clear  
d tmp(t=2)-tmp(t=1)
```

maka akan terlihat daerah-daerah yang temperaturnya lebih panas ataupun lebih dingin pada bulan Februari jika dibandingkan dengan bulan Januari (gambar 4.12).



GrADS: COLA/IGES

2008-08-28-19:45

Gambar 4.12

Perintah di atas adalah untuk menghitung perbedaan perubahan antara dua nilai yang berbeda satu selang waktu, kita juga dapat menampilkan grafik tersebut dengan perintah lainnya dengan mempergunakan offset dari waktu sekarang yaitu :

```
d tmp(t+1) - tmp
```

Jika kita membuka 2 (dua) buah file dan mempunyai model output yang sama, maka kita dapat melakukan operasi data misalnya perbedaan antara dua parameter dari dua file tersebut dengan mengetikkan :

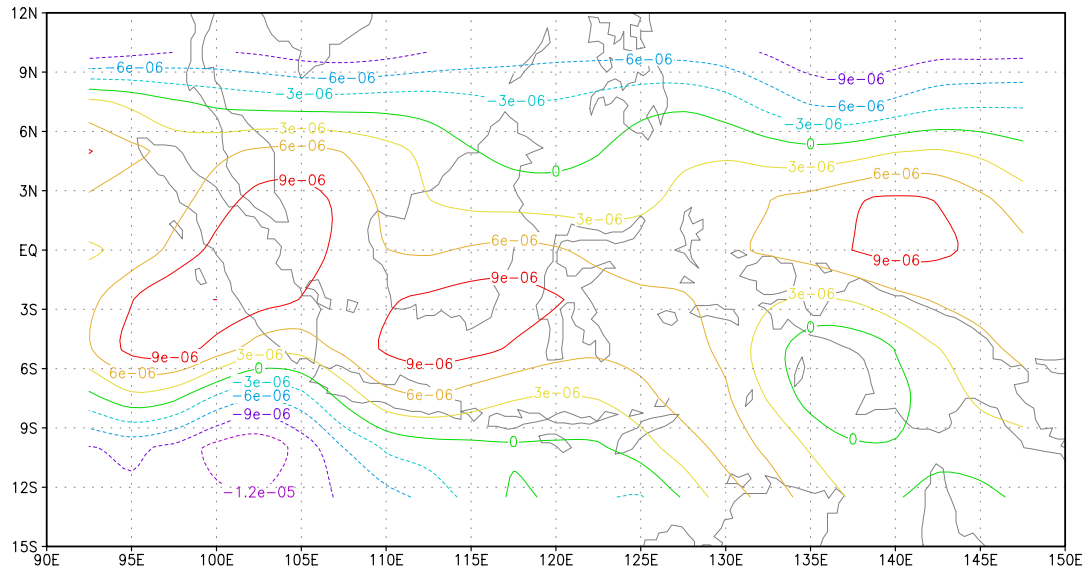
```
display tmp.2 - tmp.1
```

Perintah tersebut di atas bertujuan untuk menampilkan nilai perbedaan temperatur antara file kedua dikurangi file pertama.

Perintah lain yang berhubungan dengan menghitung nilai vortisitas relatif dengan finite differencing yaitu :

```
clear  
d hcurl(ugrd,vgrd)
```

Penghitungan komponen vertikal dari perintah `hcurl` ini pada setiap titik grid menggunakan finite differencing pada grid yang telah disediakan. Untuk syntaxnya harus mempergunakan komponen angin zonal (u) dan komponen angin meridional (v). Seperti pada gambar 4.13 yang memperlihatkan vortisitas pada lapisan 500 mb.



GrADS: COLA/IGES

2008-08-28-19:45

Gambar 4.13

Berikutnya topik kita berikutnya adalah control terhadap grafik keluaran. Salah satu perintah dari GrADS adalah `cint`, perintah ini untuk mengeset kontur interval menjadi suatu nilai tertentu, jika `cint` tidak diset maka tampilan kontur akan diset secara otomatis. Pengaturan `cint` ini akan direset dengan perintah `clear` atau `display`, lihat contoh berikut :

```
clear
set cint 1
d tmp
```

Dari perintah di atas kita akan menampilkan profil temperatur dengan interval 1 derajat.

Untuk pengaturan warna dari kontur dapat dipergunakan perintah `ccolor`, dimana syntax dari `ccolor` adalah sebagai berikut :

`set ccolor rainbow`  
`set ccolor color#`

Jika `ccolor` diset ke `rainbow` maka GrADS akan menggunakan urutan standar `rainbow` untuk menampilkan kontur individu, sedangkan kalau mempergunakan `color#` maka kita akan mempergunakan 16 warna standar dari GrADS, pilihan untuk `color#` adalah sebagai berikut :

Col#	Description	Sample	R	G	B
0	background	0	0		0 (black by default)
1	foreground	255	255		255 (white by default)
2	red	240	0		0
3	green	0	220		0
4	dark blue	31	61		250
5	light blue	0	199		199
6	magenta	220	0		99
7	yellow	230	220		51
8	orange	240	130		41
9	purple	161	0		199
10	yellow/green	161	230		51
11	medium blue	0	161		230
12	dark yellow	230	176		46
13	aqua	0	209		140
14	dark purple	110	0		220
15	gray	125	125		125

Contoh di bawah ini adalah kita menginginkan standar warna dengan `ccolor` diset 3 (green) :

```
clear
set ccolor 1
d tmp
```

Selanjutnya untuk pilihan menampilkan jenis grafik keluaran adalah dengan merubah bentuk `gxout`, syntaxnya adalah

`set gxout graphics_type`

Untuk pilihan dari `graphics_type` adalah sebagai berikut :

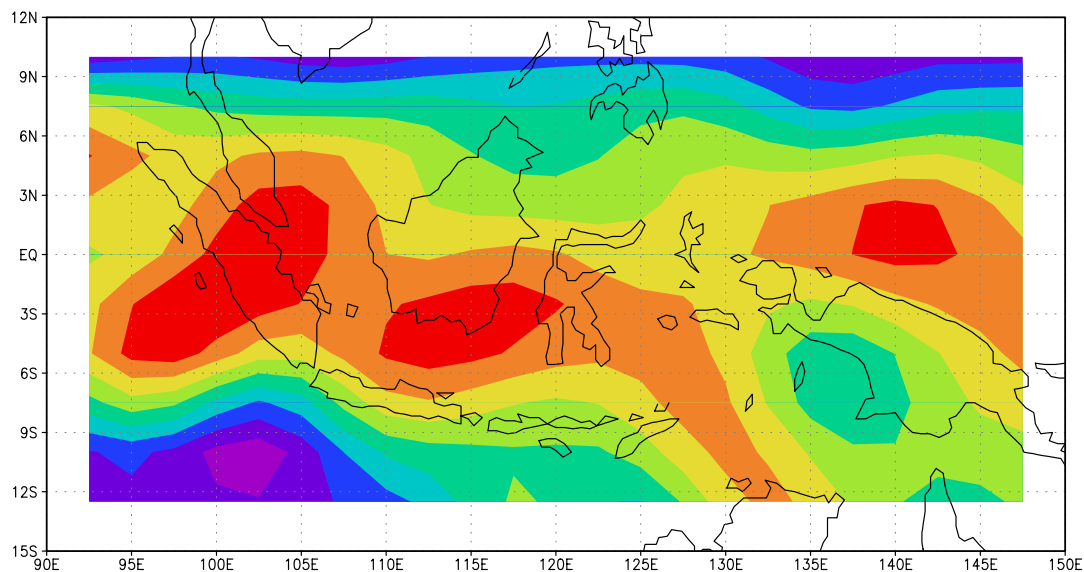
<code>bar</code>	Grafik Bar
<code>barb</code>	Plot angin dalam bentuk barb
<code>contour</code>	Plot Kontur
<code>errbar</code>	Error bars
<code>grfill</code>	Kotak Grid yang bershading
<code>fgrid</code>	Kotak Grid yang bershading dengan nilai tertentu

<code>fwrite</code>	Menyimpan dalam bentuk file biner
<code>grid</code>	Kotak Grid dengan nilai yang ditampilkan
<code>line</code>	Grafik Garis
<code>linefill</code>	Kontur dengan garis warna
<code>print</code>	Untuk menghasilkan keluaran dalam bentuk ascii
<code>scatter</code>	Grafik Scatter
<code>shaded</code>	Grafik Kontur bershading
<code>stream</code>	Grafik streamline angin
<code>stat</code>	Mencetak keluaran statistic ke suatu terminal
<code>vector</code>	Grafik vektor angin

Sekarang kita coba untuk menampilkan vortisitas angin dalam bentuk shading, ketikkan perintah berikut :

```
clear  
set gxout shaded  
d hcurl(ugrd,vgrd)
```

maka akan tampak tampilan seperti gambar 4.14 berikut :



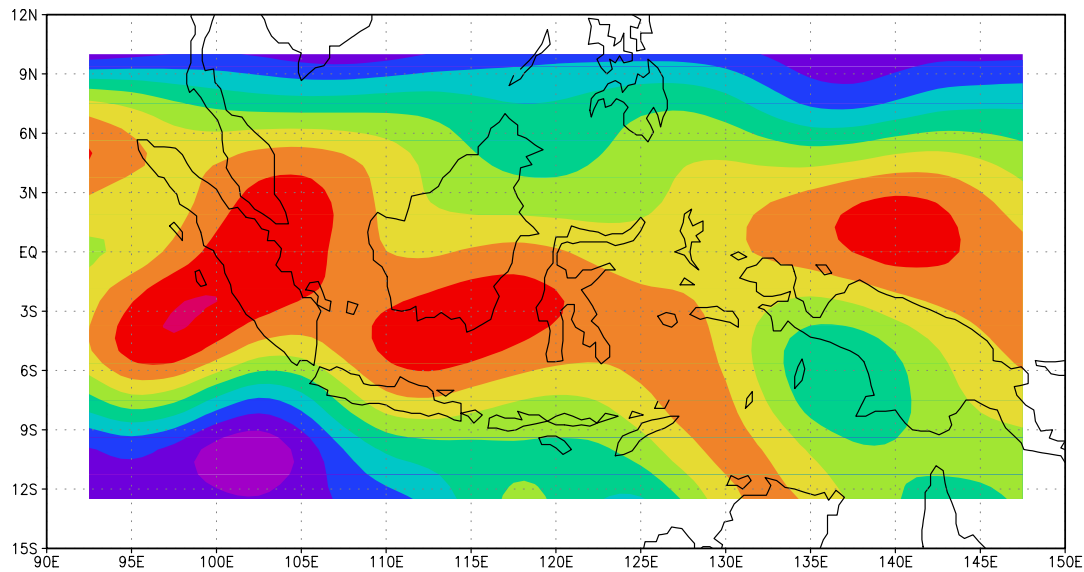
GrADS: COLA/IGES

2008-08-28-19:45

Gambar 4.14

Untuk menghasilkan kontur yang lebih halus (lihat gambar 4.15) ditambahkan perintah `csmooth`, ketikkan perintah berikut :

```
clear  
set csmooth on  
d hcurl(ugrd,vgrd)
```



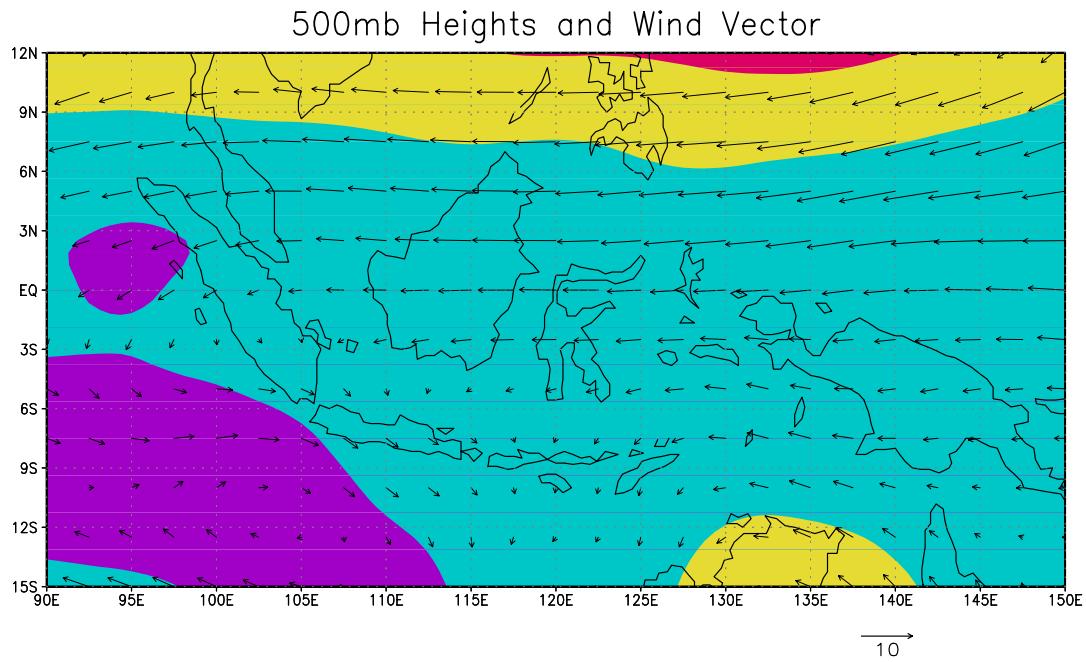
GrADS: COLA/IGES

2008-08-28-19:45

Gambar 4.15

Selanjutnya kita akan coba menggabungkan dua tipe grafik dalam satu tampilan saja, pertama kita tampilkan grafik ketinggian geopotensial dalam bentuk kontur dan grafik berikutnya adalah grafik vektor angin dalam lapisan 500 mb. Kemudian diberikan judul grafiknya. Untuk perintahnya ketikkan sebagai berikut :

```
clear  
set gxout shaded  
set ccolor 0  
set cint 10  
d hgt  
set gxout vector  
d ugrd;vgrd  
draw title 500mb Heights and Wind Vector
```



GrADS: COLA/IGES

2008-08-28-19:45

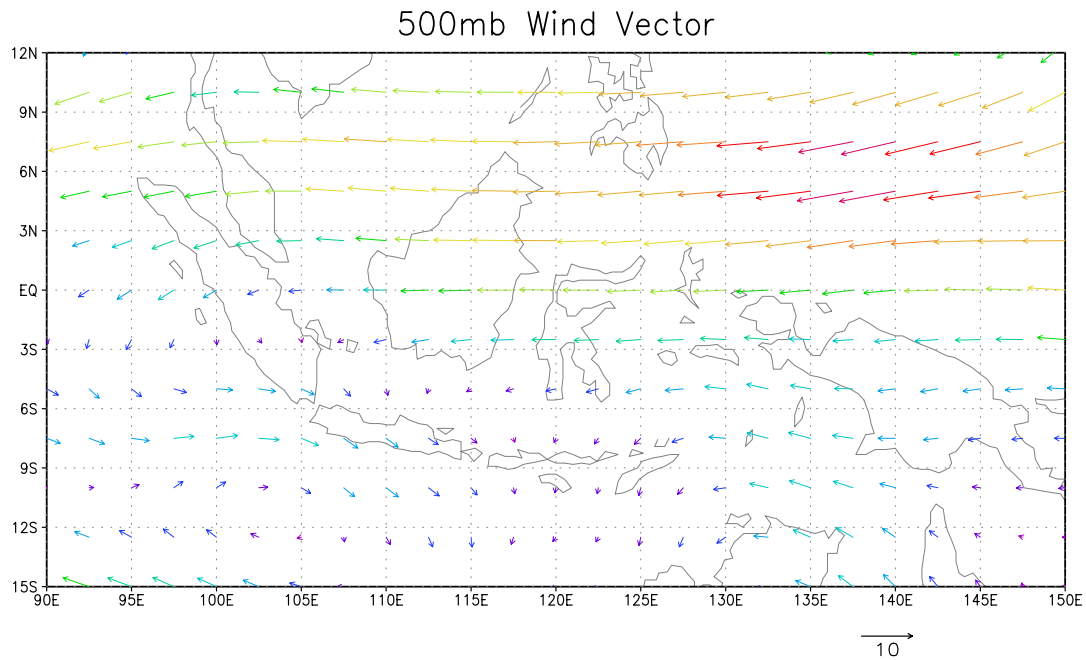
Gambar 4.16

Grafik berikut ini adalah grafik yang menampilkan vektor angin dan vektor tersebut diwarnai dengan magnitudo angin

```
clear  
d ugrd;vgrd;mag(ugrd,vgrd)  
draw title 500mb Wind Vector
```

Maka akan terlihat gambar vektor angin yang disertai dengan warna yang menggambarkan magnitudo angin (gambar 4.17).





GrADS: COLA/IGES

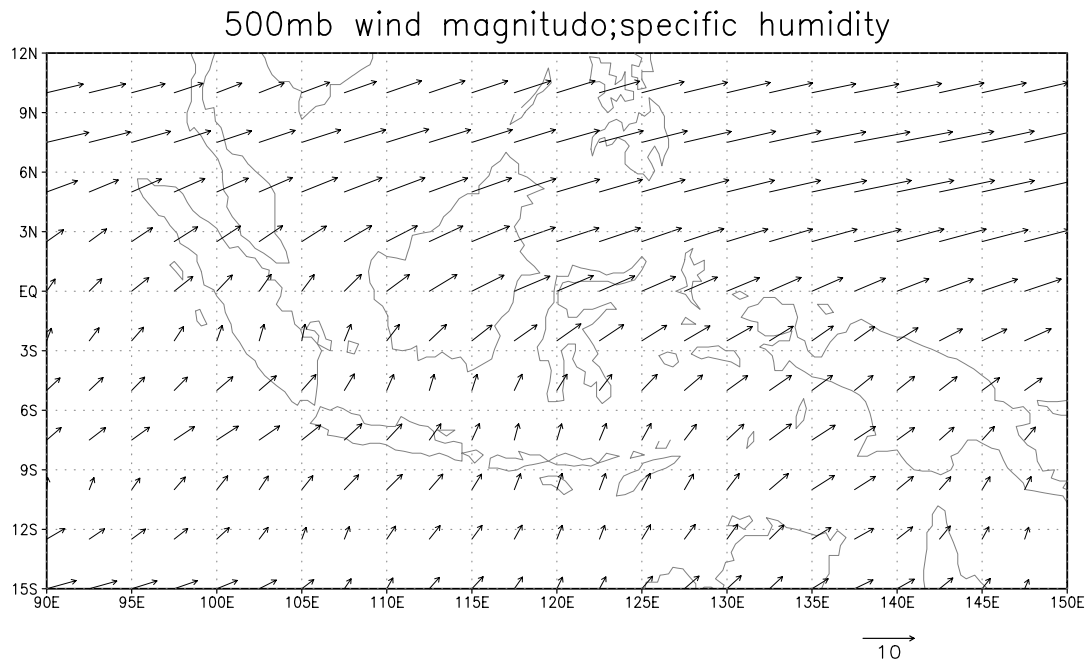
2008-08-28-19:45

Gambar 4.17

Kita juga dapat menampilkan pseudo vektor dengan menampilkan parameter yang kita inginkan, misalkan untuk komponen U adalah kecepatan angin sedangkan komponen V adalah kelembaban spesifik :

```
clear  
d mag(ugrd,vgrd);spf*10000  
draw title 500mb wind magnitudo;specific humidity
```

Perlu diingat : Gambar 4.18 di bawah ini bukan menunjukkan gambar vektor angin tetapi plot antara magnitudo angin dan kelembaban spesifik



GrADS: COLA/IGES

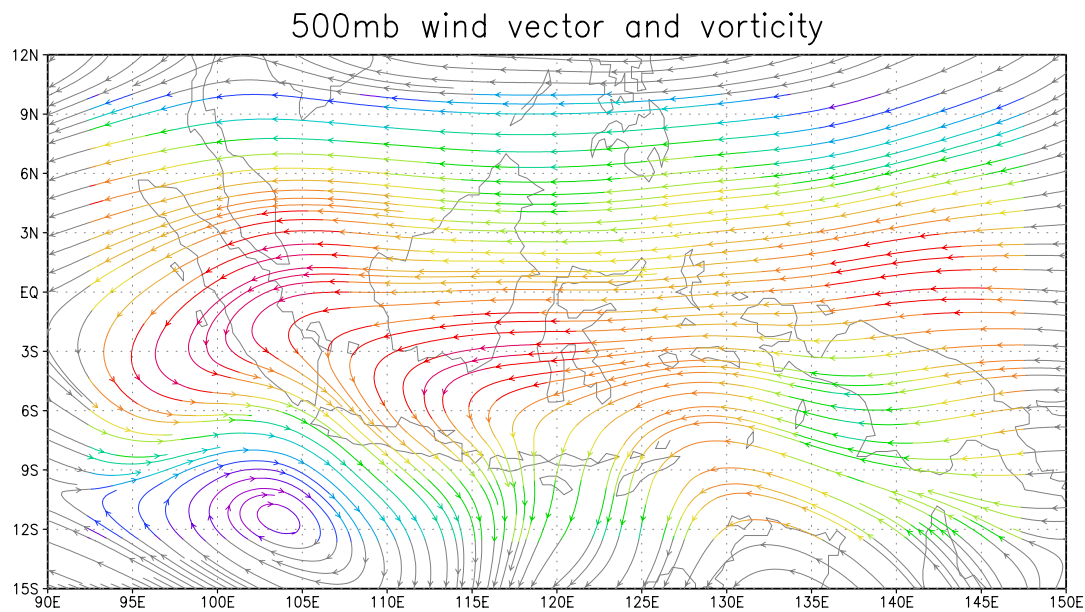
2008-08-28-20:31

Gambar 4.18

Selanjutnya dicoba untuk menampilkan streamline angin dengan diwarnai vortisitas angin, ketikkan perintah sebagai berikut :

```
clear  
set gxout stream  
d ugrd;vgrd;hcurl(ugrd,vgrd)  
draw title 500mb wind vector and vorticity
```

Gambar 4.19 berikut ini adalah salah satu bentuk penampilan plot angin tetapi dalam bentuk streamline.



GrADS: COLA/IGES

2008-08-28-19:45

Gambar 4.19

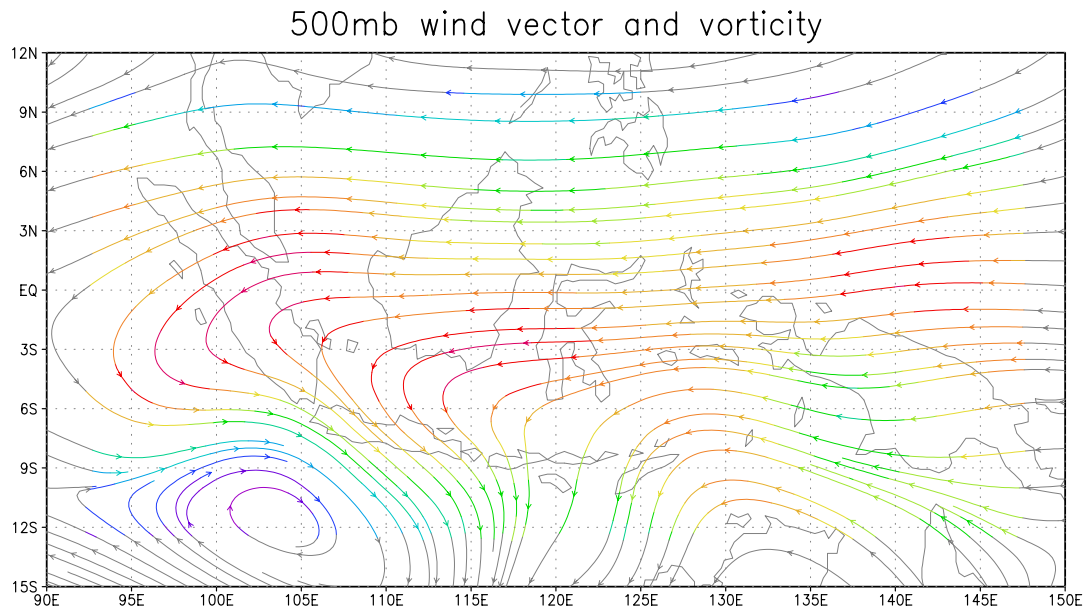
Untuk mengatur kerapatan dari garis-garis streamline dapat digunakan perintah `strmden`, yang syntaxnya adalah :

`set strmden value`

Dimana kerapatan dari streamline tersebut berkisar dari 1 – 10 dan defaultnya adalah 5. Jika kita ulangi perintah di atas dengan merubah kerapatan menjadi 3 maka akan dihasilkan grafik sebagai berikut :

```
clear
set gxout stream
set strmden 3
d ugrd;vgrd;hcurl(ugrd,vgrd)
draw title 500mb wind vector and vorticity
```

Pada gambar 4.20 di bawah terlihat bahwa streamline yang ditampilkan mempunyai kerapatan yang lebih kurang jika dibandingkan dengan gambar 4.19 di atas.



GrADS: COLA/IGES

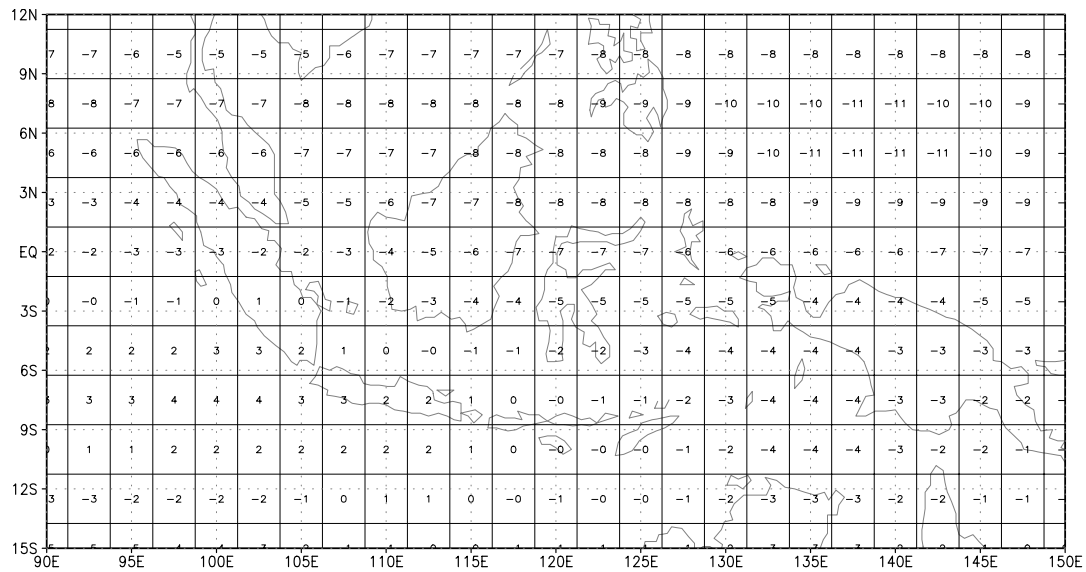
2008-08-28-19:45

Gambar 4.20

Kalau ingin menampilkan nilai aktual dari setiap grid maka penggunaan keluaran berbentuk grid sebagai berikut :

```
clear  
set gxout grid  
d ugrd
```

Terlihat bahwa setiap kotak pada grid tampil nilai dari angin zonalnya (lihat gambar 4.21).



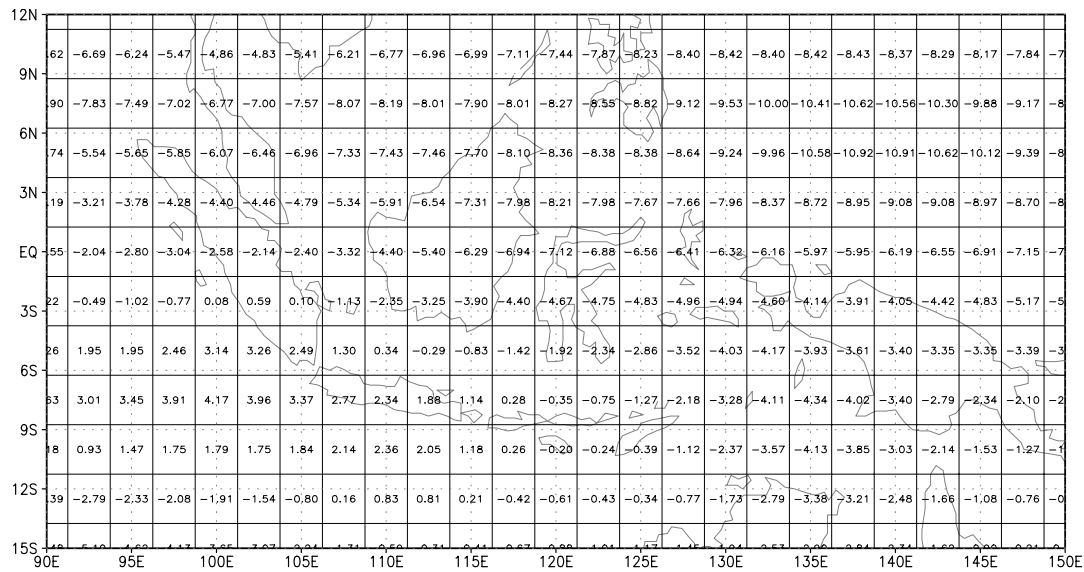
GrADS: COLA/IGES

2008-08-28-19:45

Gambar 4.21

Jika kita ingin menampilkan angka desimalnya maka penggunaan perintah `dignum` size yang biasanya sebelumnya diikuti oleh perintah `digsiz` yang berfungsi untuk menentukan ukuran string.

```
clear  
set digsiz 0.07  
set dignum 2  
d ugrd
```



GrADS: COLA/IGES

2008-08-28-19:45

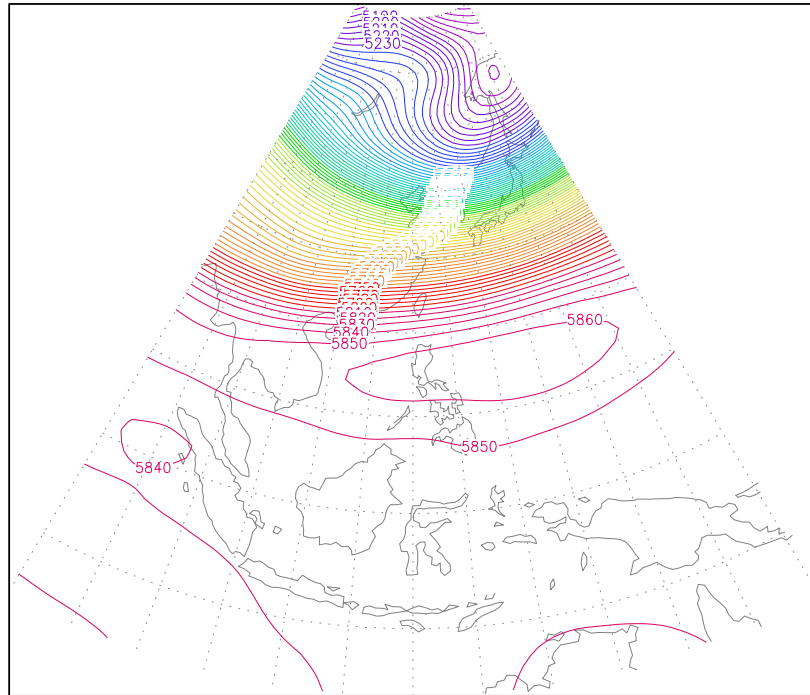
Gambar 4.22

Sekarang bagaimana cara menampilkan dalam plot polar stereographic, dalam kasus ini kita ingin GrADS mengoperasikan data dari bujur 140 BB – 40 BB dan lintang 15 LU – 80 LU

Untuk menggambar koordinat polar tersebut ketikkan perintah berikut :

```

clear
set lon 90 150
set lat -15 70
set mproj nps
set gxout contour
set cint 10
d hgt
    
```



GrADS: COLA/IGES

2008-08-28-19:45

Gambar 4.23

Untuk mengeset jenis-jenis proyeksi peta terdapat beberapa pilihan berikut :

<b>latlon</b>	Proyeksi terhadap bujur dan lintang (default)
<b>scaled</b>	Rasio terhadap bujur/lintang tidak dipertahankan
<b>nps</b>	polar stereographic utara
<b>sps</b>	polar stereographic selatan
<b>lambert</b>	proyeksi Lambert conformal conic
<b>mollweide</b>	proyeksi Mollweide
<b>orthogr</b>	proyeksi Orthographic
<b>robinson</b>	proyeksi Robinson
<b>off</b>	Tidak menampilkan peta; label axis tidak diinterpretasikan sebagai bujur/lintang

Salah satu kemampuan dari GrADS adalah dapat membuat beberapa keluaran dalam bentuk file gambar. Format gambar yang sering dipergunakan yaitu yang berekstension .gif, .gmf, dan .eps. Format-format gambar tersebut kita sesuaikan dengan keperluan pengolahan selanjutnya. Sekarang kita akan membahas beberapa perintah GrADS yang dapat membuat file gambar tersebut.

- Perintah **PRINTIM**

Perintah `printim` ini akan menghasilkan image dengan format berekstension `.PNG` atau `.GIF`, dimana syntaxnya adalah sebagai berikut :

`printim filename options`

dimana

*filename*: Nama dari file keluaran

*options*: `gif` - menghasilkan keluaran format gif (defaultnya `.png`)  
`black` - latar belakang hitam (default)  
`white` - latar belakang putih  
`xNNN` - ukuran mendatar dalam `NNN` piksel  
`yNNN` - ukuran vertikal size dalam `NNN` piksel

- Perintah **PRINT**

Perintah `print` ini adalah untuk membuat suatu salinan tampilan GrADS menjadi suatu file dalam format metacode.

Dimana urutan penggunaan perintah `print` ini adalah sebagai berikut :

```
enable print fname % fname diisi dengan lokasi dan
                    nama file, umumnya ekstension
                    yang diberikan adalah .gmf
print                % mencetak tampilan menjadi
                    format metacode
disable print        % menonaktifkan kembali perintah
                    print
```

Apabila kita menggunakan perintah `print` ini maka akan terdapat suatu file image di lokasi yang telah kita tentukan sebelumnya dan mempunyai ekstension `.gmf` atau `.gx` (penamaan ekstension tergantung oleh user) dan untuk membuka file image tersebut dibutuhkan suatu *software* viewer yaitu `gv32.exe`. Software `gv32.exe` ini merupakan *software* khusus yang disertakan oleh GrADS yang berfungsi untuk menampilkan dan memanipulasi GrADS metafile. Salah satu kelebihan dari software ini adalah kita dapat memperbesar gambar tanpa membuat gambar tersebut pecah seperti halnya file yang berekstension `.gif` dikarenakan formatnya sudah dalam bentuk vektor.

- Perintah **!gxeps**



Perintah `!gxeps` ini berfungsi untuk mengkonversi format file metacode menjadi sebuah file PostScript, adapun syntaxnya adalah :

```
gxeps [ -acdilrsv -i <infile> -o <outfile>  ] [<infile>] ]
```

Jika kita tidak memberikan opsi kepada perintah `!gxeps` ini maka perintah `!gxeps` ini akan mengkonversi file format metacode ke format greyscale dan dengan latar belakang putih pada file PostScriptnya.

Berikut ini adalah opsi pada perintah `gxeps` :

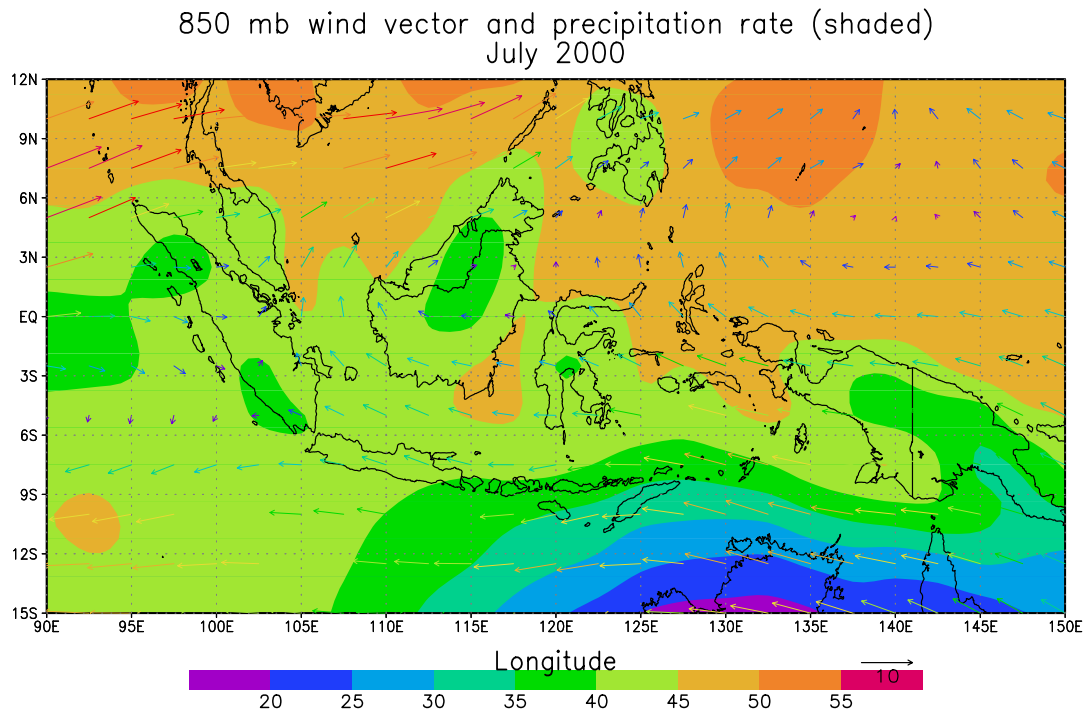
- `-i fname`     nama input file GrADS metacode
- `-o fname`     nama file output PostScript
- `-c`             print berwarna pada latar belakang putih
- `-r`             print greyscale pada latar belakang hitam
- `-d`             menambahkan CTRL-D pada akhir file berguna jika mencetak pada printer HP 1200C/PS color printer.
- `-1`             menggunakan *PostScript Level 1*.
- `-2`             menggunakan *PostScript Level 2*.
- `-a`             Ukuran kertas *DIN A4*.
- `-l`             Ukuran kertas *US-Letter*.
- `-L`             Untuk menanyakan sebuah untuk dicetak pada grafik
- `-n`             Untuk menanyakan sebuah catatab yang terdapat pada header file PostScript
- `-s`             Untuk menambahkan sebuah file dan tampilan waktu pada grafik
- `-v`             mode verbose

Untuk membuka file PostScript ini dibutuhkan software yang bernama GhostView.

Berikut ini adalah contoh perintah untuk membuat suatu plot vektor angin dan precipitation rate (shaded) di wilayah Indonesia, tampilan grafik tersebut kemudian dikonversi menjadi file berekstension `.gif`, `.gmf` dan `.eps` (gambar 4.24). Adapun perintahnya adalah sebagai berikut :



```
reinit
*buka file
open h:/grads/ctl/all.prs
open h:/grads/ctl/pwat.sfc
*setting lingkungan
set lon 90 150
set lat -15 12
set lev 850
set t 7
set mpdset hires
set csmooth on
*plot precipitation rate
set gxout shaded
set clevs 20 25 30 35 40 45 50 55
d pwat.2
cbar
*plot vektor angin
set gxout vector
d ugrd.1;vgrd.1;mag(ugrd.1,vgrd.1)
*membuat judul
draw title 850 mb wind vector and precipitation rate (shaded)\July 2000
draw xlab Longitude
draw ylab Latitude
*cetak ke file berekstension .gif
printim h:/grads/image/windpw.gif white
*cetak ke file GrADS metacode
enable print h:/grads/image/windpw.gmf
print
disable print
*cetak ke file PostScript
!gxeps -c -i h:/grads/image/windpw.gmf -o h:/grads/image/windpw.eps
```



GrADS: COLA/IGES

2008-08-28-19:45

Gambar 4.24

Selanjutnya kita akan membahas tampilan GrADS yang dapat memuat beberapa tampilan dalam satu layar saja, untuk melakukan hal ini dalam dipergunakan perintah `set parea`, syntaxnya adalah sebagai berikut :

```
set parea xmin xmax ymin ymax
```

dimana `xmin` adalah posisi awal x (0-10.5)

`xmax` adalah posisi akhir x (0-10.5)

`ymin` adalah posisi awal y (0-8)

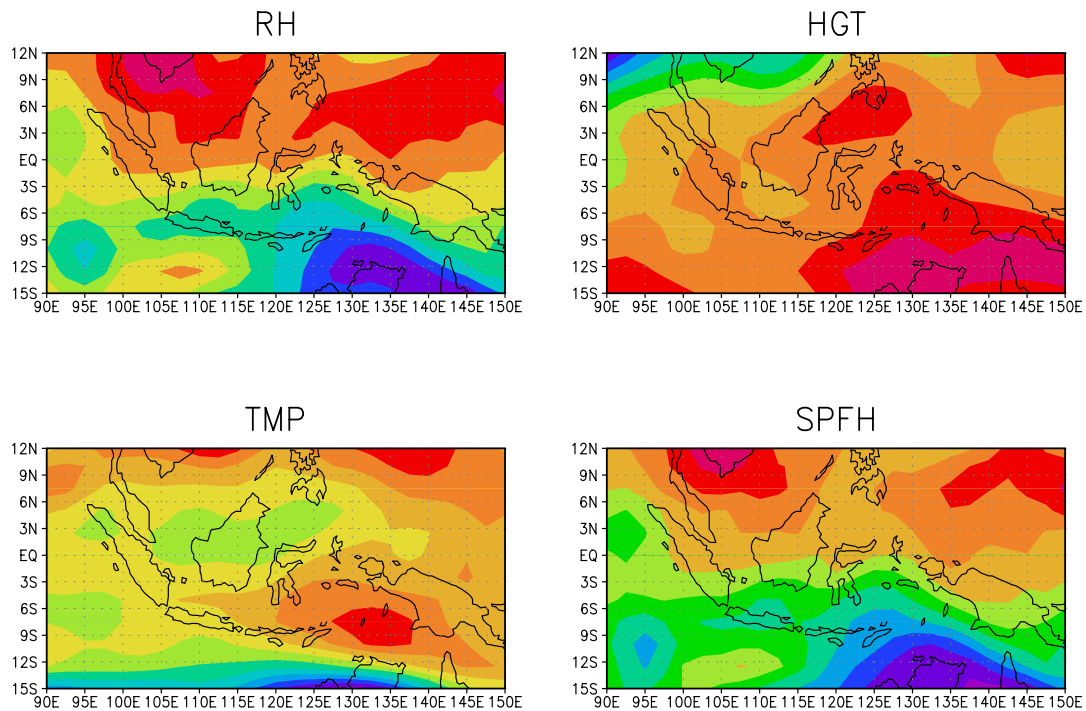
`ymax` adalah posisi akhir y (0-8)

Sebagai contoh kita ingin membuat tampilan RH, HGT, TMP dan SPFH dalam tampilan satu halaman saja, maka untuk perintahnya ketikkan perintah sebagai berikut

```
reinit  
open h:/grads/ctl/all.prs  
set lon 90 150  
set lat -15 12  
set t 8
```

```
set lev 700
set gxout shaded
*RH
set parea 0.5 5 5 8
d rh
draw title RH
*HGT
set parea 6 10.5 5 8
d hgt
draw title HGT
*TMP
set parea 0.5 5 1 4
d tmp
draw title TMP
*SPFH
set parea 6 10.5 1 4
d spfh
draw title SPFH
```

Maka akan tampilannya adalah seperti pada gambar 4.25 berikut ini :



Gambar 4.25

## Menyalin Data menjadi file Biner

Kadangkala kita hanya membutuhkan suatu data biner hanya pada grid tertentu saja dan tidak memerlukan data global, maka kita dapat mengambil data yang kita ingin tersebut hanya di lokasi yang kita perlukan saja dengan menjadikan file biner baru yang berekstension fwrite.

Misalnya kita hanya ingin mengambil data temperature di lintang 0 derajat dan bujur 115°BT selama 12 (dua belas) bulan, maka dapat kita lakukan dengan menjalankan urutan perintah sebagai berikut :

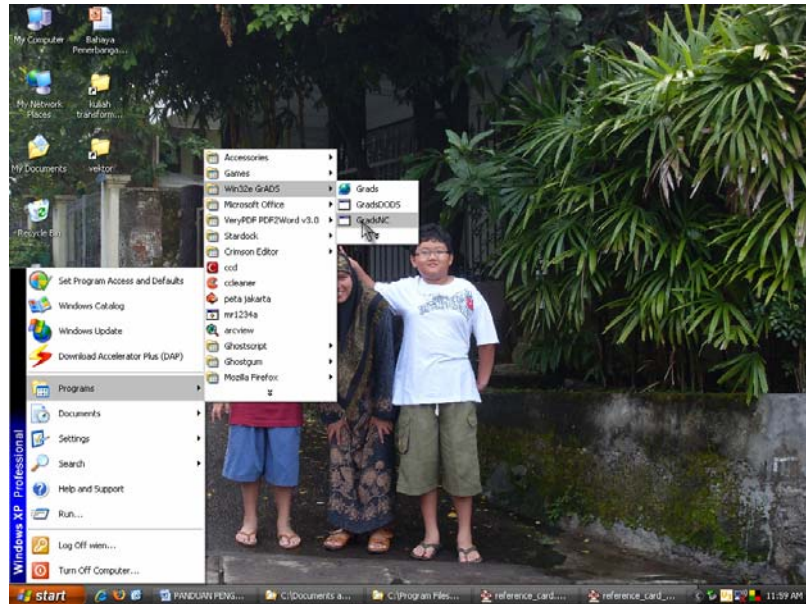
```
ga- >open h:/grads/ctl/all.prs
ga- >set lat 0
ga- >set lon 115
ga- >set t 1 12
ga- >set gxout fwrite
ga- >set fwrite h:/grads/zonal.fwrite
ga- >d ugrd
ga- >disable fwrite
```

Maka kita akan suatu file baru yang bernama zonal.fwrite yang hanya berisi data angin zonal di lintang dan bujur yang telah kita tentukan tadi

## Perintah GrADS untuk membuka file netCDF

Khusus untuk membuka file yang berbasis netCDF yang umumnya berekstension .nc maka kita hanya membutuhkan satu file saja tidak seperti waktu membuka file berformat biner yang memerlukan tiga file (file descriptor, file index, dan file data).

Untuk membuka file tersebut kita harus mengaktifkan program GradsNC dari start menu seperti pada gambar 4.26 berikut ini :



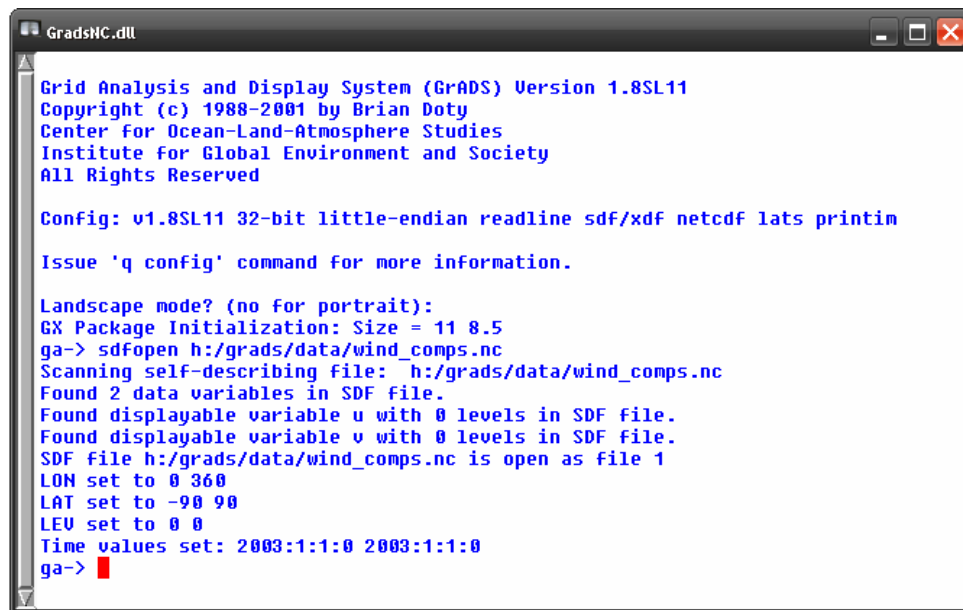
Gambar 4.26

Selanjutnya akan muncul prompt yang sama dengan pada saat kita membuka program grads seperti yang telah dijelaskan di awal.

Dari prompt ketikkan perintah berikut ini :

```
ga- >sdfopen h:/grads/data/wind_comps.nc
```

maka akan muncul deskripsi file seperti pada gambar 4.27 berikut ini



```
GradsNC.dll
Grid Analysis and Display System (GrADS) Version 1.8SL11
Copyright (c) 1988-2001 by Brian Doty
Center for Ocean-Land-Atmosphere Studies
Institute for Global Environment and Society
All Rights Reserved

Config: v1.8SL11 32-bit little-endian readline sdf/xdf netcdf lats printin

Issue 'q config' command for more information.

Landscape mode? (no for portrait):
GX Package Initialization: Size = 11 8.5
ga-> sdfopen h:/grads/data/wind_comps.nc
Scanning self-describing file: h:/grads/data/wind_comps.nc
Found 2 data variables in SDF file.
Found displayable variable u with 0 levels in SDF file.
Found displayable variable v with 0 levels in SDF file.
SDF file h:/grads/data/wind_comps.nc is open as file 1
LON set to 0 360
LAT set to -90 90
LEU set to 0 0
Time values set: 2003:1:1:0 2003:1:1:0
ga-> █
```

Gambar 4.27

Apabila muncul penjelasan seperti di atas maka file tersebut telah terbuka dan siap untuk kita olah sebagaimana perintah-perintah yang telah dijelaskan sebelumnya. File data netCDF tersebut terdiri dari dua variabel saja yaitu u dan v (angin zonal dan meridional).



## Apa itu GrADS Data Server ?

Grads Data Server (GDS, yang dikenal sebagai GrADS-DODS Server) adalah data server yang dilindungi yang menyediakan subsetting dan pelayanan analisis melalui internet.

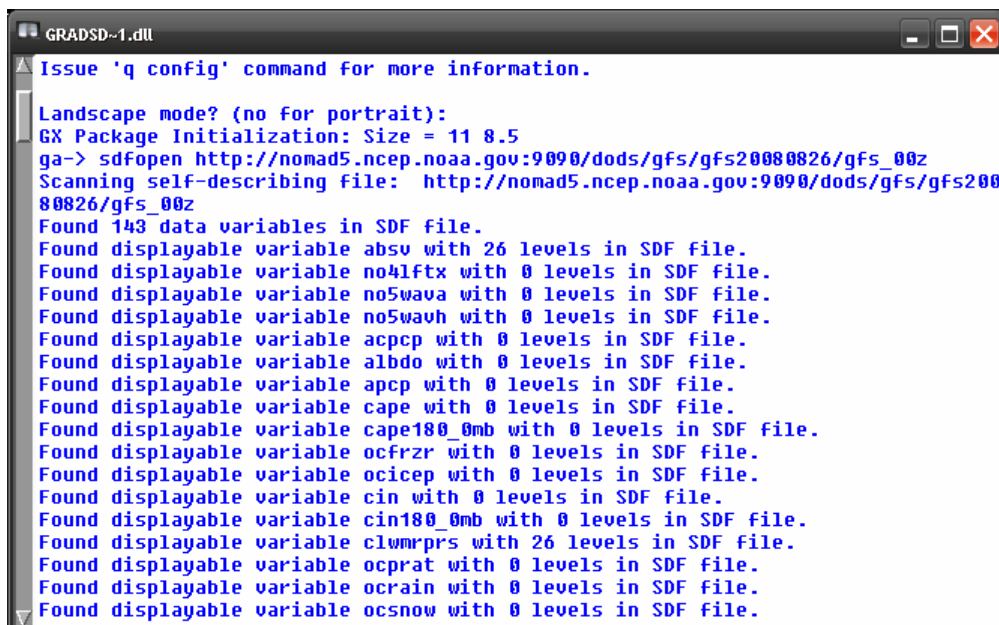
Untuk mengakses data tersebut, GrADS menyediakan beberapa server yang dapat di akses melalui website GrADS, yaitu melalui alamat : <http://www.iges.org/grads/gds/> dimana pada alamat tersebut terdapat banyak institusi atau lembaga yang menyediakan data-data global maupun regional baik reanalisis maupun forecasting. Kita bisa mengakses data tersebut dengan mengaktifkan GradsDODS.

Kemudian dari prompt GrADS ketikkan alamat data yang ingin kita akses, misalnya pada alamat : [http://nomad5.ncep.noaa.gov:9090/dods/gfs/gfs20080826/gfs\\_00z](http://nomad5.ncep.noaa.gov:9090/dods/gfs/gfs20080826/gfs_00z) yang berisi file Global Forecast System (GFS) yang terdiri dari 43 variabel meteorologi yang merupakan prakiraan dari 00Z26AUG2008 sampai dengan 12Z02SEP2008 (dengan resolusi setiap 3 jam).

Dari prompt GrADS ketikkan perintah sebagai berikut :

```
ga->sdfopen http://nomad5.ncep.noaa.gov:9090/dods/gfs/gfs20080826/gfs_00z  
(enter)
```

maka akan muncul penjelasan seperti pada gambar 4.28 berikut :



```
GRADSD-1.dll  
Issue 'q config' command for more information.  
Landscape mode? (no for portrait):  
GX Package Initialization: Size = 11 8.5  
ga-> sdfopen http://nomad5.ncep.noaa.gov:9090/dods/gfs/gfs20080826/gfs_00z  
Scanning self-describing file: http://nomad5.ncep.noaa.gov:9090/dods/gfs/gfs20080826/gfs_00z  
Found 143 data variables in SDF file.  
Found displayable variable absv with 26 levels in SDF file.  
Found displayable variable no4lftx with 0 levels in SDF file.  
Found displayable variable no5wava with 0 levels in SDF file.  
Found displayable variable no5wavh with 0 levels in SDF file.  
Found displayable variable acpcp with 0 levels in SDF file.  
Found displayable variable albdo with 0 levels in SDF file.  
Found displayable variable apcp with 0 levels in SDF file.  
Found displayable variable cape with 0 levels in SDF file.  
Found displayable variable cape180_0mb with 0 levels in SDF file.  
Found displayable variable ocfrzr with 0 levels in SDF file.  
Found displayable variable ocicep with 0 levels in SDF file.  
Found displayable variable cin with 0 levels in SDF file.  
Found displayable variable cin180_0mb with 0 levels in SDF file.  
Found displayable variable clwmprsr with 26 levels in SDF file.  
Found displayable variable ocprat with 0 levels in SDF file.  
Found displayable variable ocrcrain with 0 levels in SDF file.  
Found displayable variable ocsnow with 0 levels in SDF file.
```

Gambar 4.28

Setelah itu kita dapat menampilkan apa yang kita inginkan sebagaimana yang telah dijelaskan di atas.

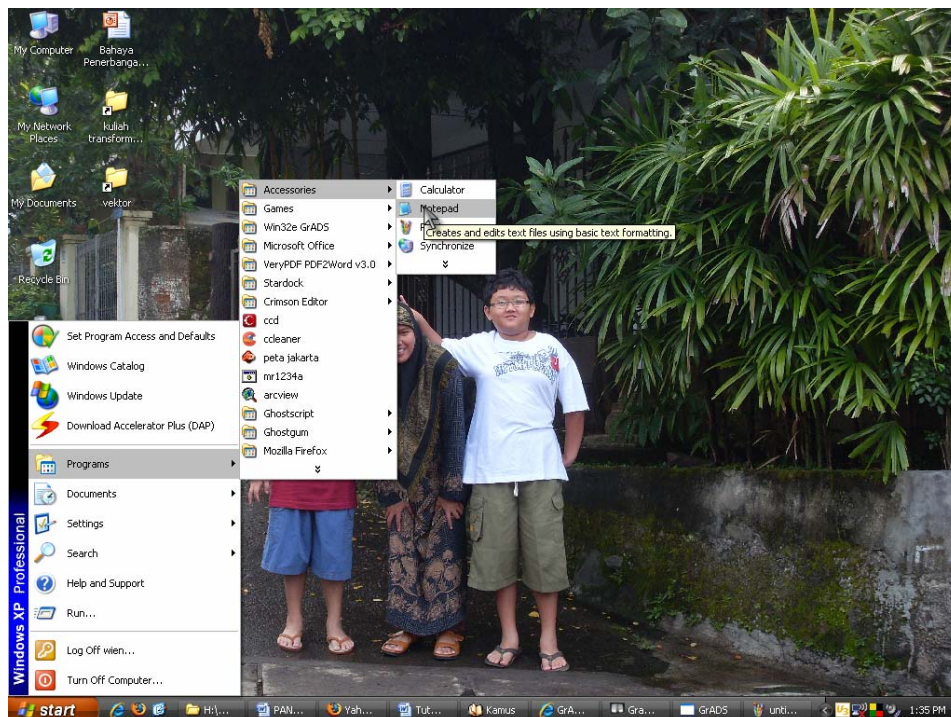
## BAB V

### PENGGUNAAN SCRIPT PADA GrADS

Untuk menampilkan suatu grafik melalui GrADS maka kita dapat mengetikkan perintahnya langsung pada layar perintah GrADS, tetapi kadangkali kita harus mengetikkan perintah yang sangat banyak. Untuk mengulang perintah yang telah diketikkan kita dapat menggunakan tombol anak panah keatas, menekannya berulang kali sampai menemukan perintah yang kita maksud dan setelah itu dapat diedit. Kadangkala untuk perintah yang banyak sekali kita mengalami kerepotan diakibatkan banyaknya perintah tersebut dan juga urutan-urutannya. Kalau kita salah mengetikkan urutan perintah dapat berakibat gambar yang ditampilkan juga tidak seperti yang kita harapkan. Solusi dari permasalahan tersebut adalah dengan program menggunakan script. Dengan sekali menjalankan script maka kita dapat mengontrol urutan perintah serta mengedit program dengan cara yang lebih cepat. Untuk membuat script GrADS sangatlah mudah, kita dapat mempergunakan beberapa program pengolah kata, tetapi disini kita akan mempergunakan program notepad.

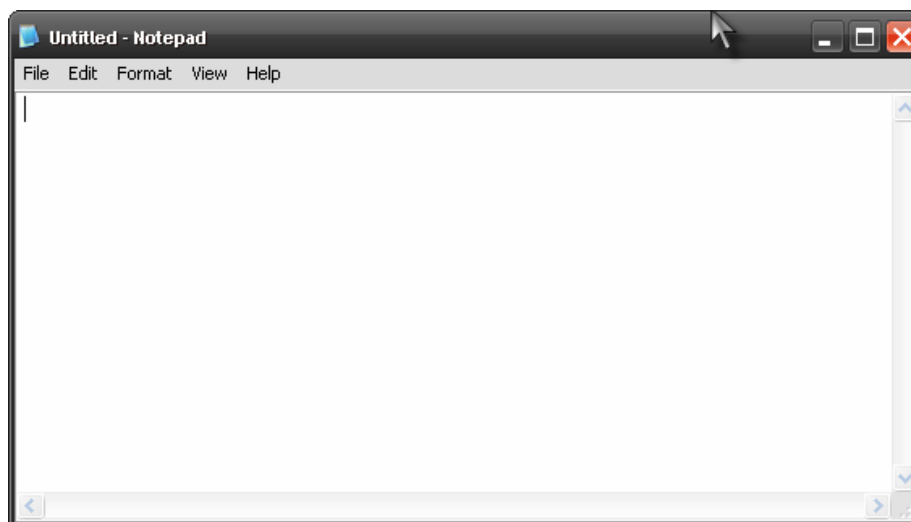
#### 5.1 Urutan pembuatan script GrADS.

- a. Buka program notepad dari menu windows (lihat gambar 5.1)



Gambar 5.1

- b. Setelah klik ikon notepad maka akan muncul layar pengetikan seperti pada gambar 5.2 berikut ini :



Gambar 5.2

- c. Setelah muncul layar tersebut kita dapat langsung mengetikkan perintah. Misalkan kita ingin menampilkan grafik vektor angin, maka ketikkan perintah di layar GrADS mulai dari membuka file descriptor sampai dengan mencetak grafik menjadi file berekstension .gif



Gambar 5.3

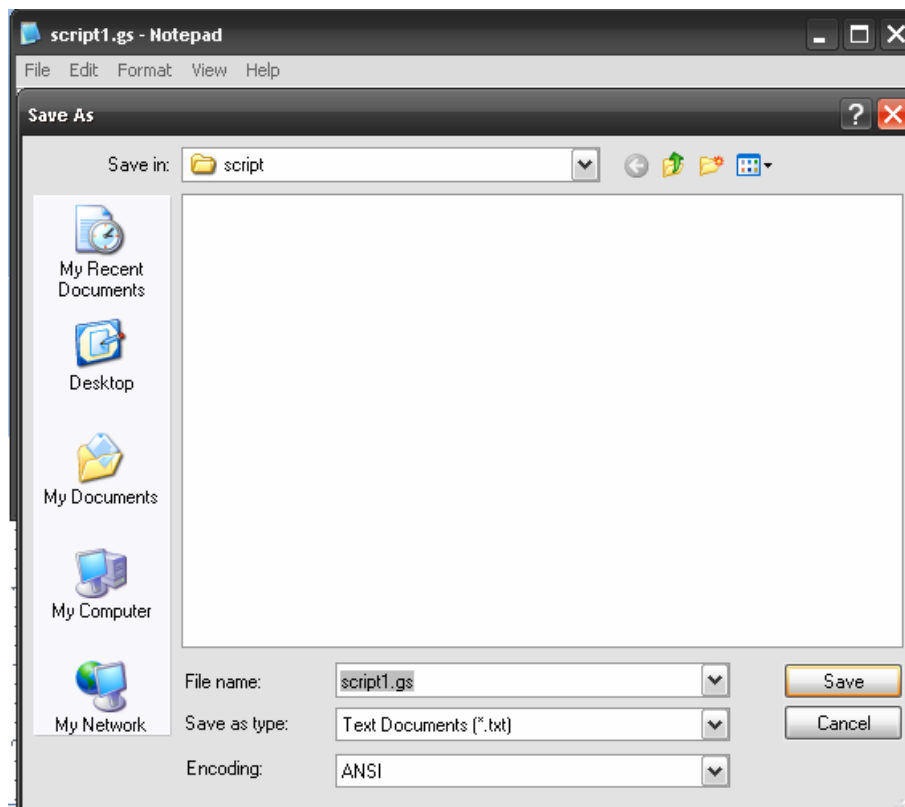
### Penjelasan perintah

**'reinit'** % menutup semua file dan mengembalikan setting ke setting awal

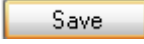
```
'open h:/grads/ctl/all.prs'      % membuka file descriptor all.prs
'set lon 90 150'                % setting bujur 90BT-150BT
'set lat -15 12'                % setting lintang 15 LS – 12LU
'set t 1'                       % setting waktu ke bulan Januari
'set lev 1000'                  % setting ketinggian 1000 mb
'set gxout vector'              % setting tampilan ke vektor
'set mpdset hires'              % untuk menampilkan peta pada resolusi tinggi
'set csmooth on'                % untuk menampilkan kontur lebih halus
'd ugrd;vgrd'                   % untuk menampilkan vektor angin
'draw title Vektor Angin di Wilayah Indonesia\Bulan Januari 2000 ketinggian
                               1000 mb'          % membuat judul
'draw xlab bujur'                % membuat label untuk x axis
'draw ylab lintang'              % membuat label untuk y axis
'printim h:/grads/image/vektor_angin_jan00.gif white' % mencetak grafik
                               dalam bentuk file berekstension .gif dan berlatar
                               belakang putih.
```

Perlu diingat : bahwa terdapat sedikit perbedaan antara cara penulisan pada layar perintah GrADS dan penulisan script, dimana dalam script perintahnya harus berada di dalam tanda petik '*ekspresi GrADS*'. Misalnya pada layar kita ketikkan perintah *set gxout shaded* maka di script kita ketikkan '*set gxout shaded*'

Kalau selesai mengetikkan perintah tersebut maka jangan lupa disimpan di lokasi yang kita telah tentukan, misalkan kita simpan dengan nama script1.gs (untuk diingat: penulisan ekstension file script harus berakhiran .gs) di alamat misalnya h:/grads/script seperti pada gambar 5.4 berikut :



Gambar 5.4

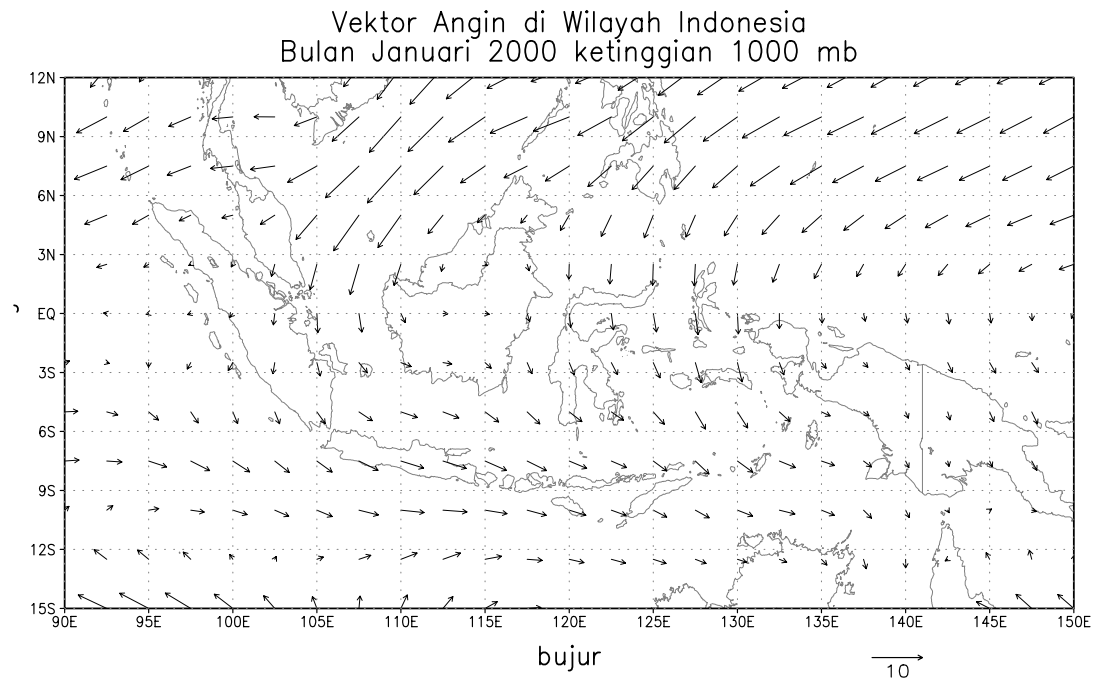
Klik tombol , maka kita siap untuk menjalankan script tersebut.

## 2.2 Menjalankan Script GrADS

- a. Buka layar GrADS
- b. Setelah layar terbuka, pada prompt ketikkan perintah :

```
ga- >run h:/grads/script/script1.gs (enter)
```

- c. Jika tidak ada kesalahan pada pengetikan script maka GrADS akan menjalankan urutan-urutan perintah yang telah diketikkan. Jika terdapat kesalahan maka GrADS akan menampilkan jenis kesalahan. Kesalahan ketik tersebut dapat diedit dari layar notepad dan jangan lupa disimpan kembali. Setelah diperbaiki, script tersebut dapat dijalankan kembali. Jika program tersebut dijalankan maka akan muncul grafik seperti pada gambar 5.5 berikut :



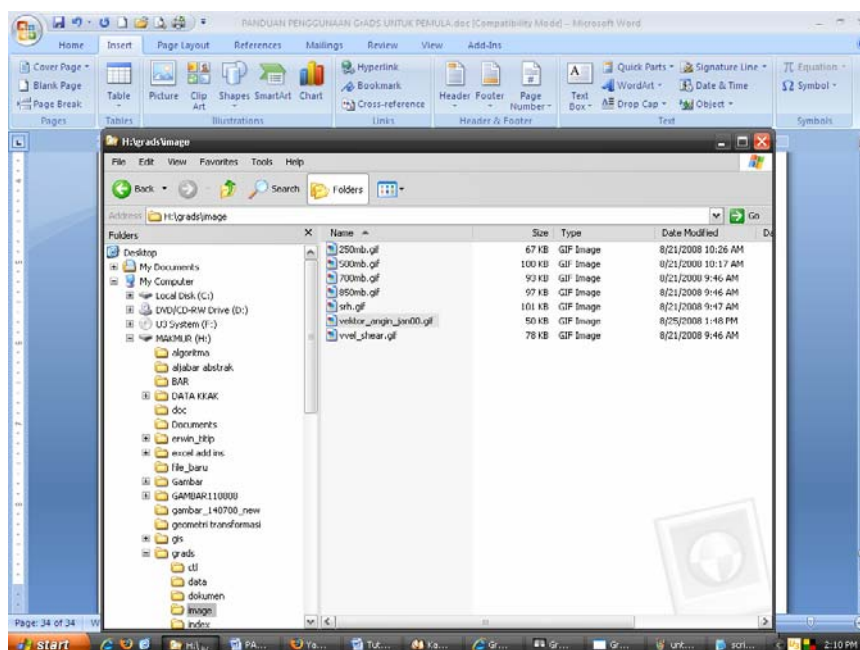
GrADS: COLA/IGES

2008-08-28-19:47

Gambar 5.5

- d. Selain menampilkan grafik pada layar GrADS, kita juga membuat sebuah file gambar berekstension .gif yang bernama vektor\_angin\_jan00.gif pada lokasi h:/grads/image/. Jika GrADS menjalankan perintah tersebut dengan sempurna maka akan terbuat file tersebut pada lokasi tersebut. File tersebut berikutnya dapat kita olah kembali tergantung dari keperluan kita.

Catatan : dalam penulisan nama file dihindari adanya spasi di antara nama file karena GrADS tidak membolehkan hal tersebut. Misalnya h:/grads/image/vektor angin jan00.gif, penulisan tersebut salah dikarenakan adanya spasi, penulisan yang benar adalah h:/grads/image/vektor\_angin\_jan00.gif



Gambar 5.6

### 2.3 Beberapa contoh script GrADS yang biasa dipergunakan dalam bidang Meteorologi

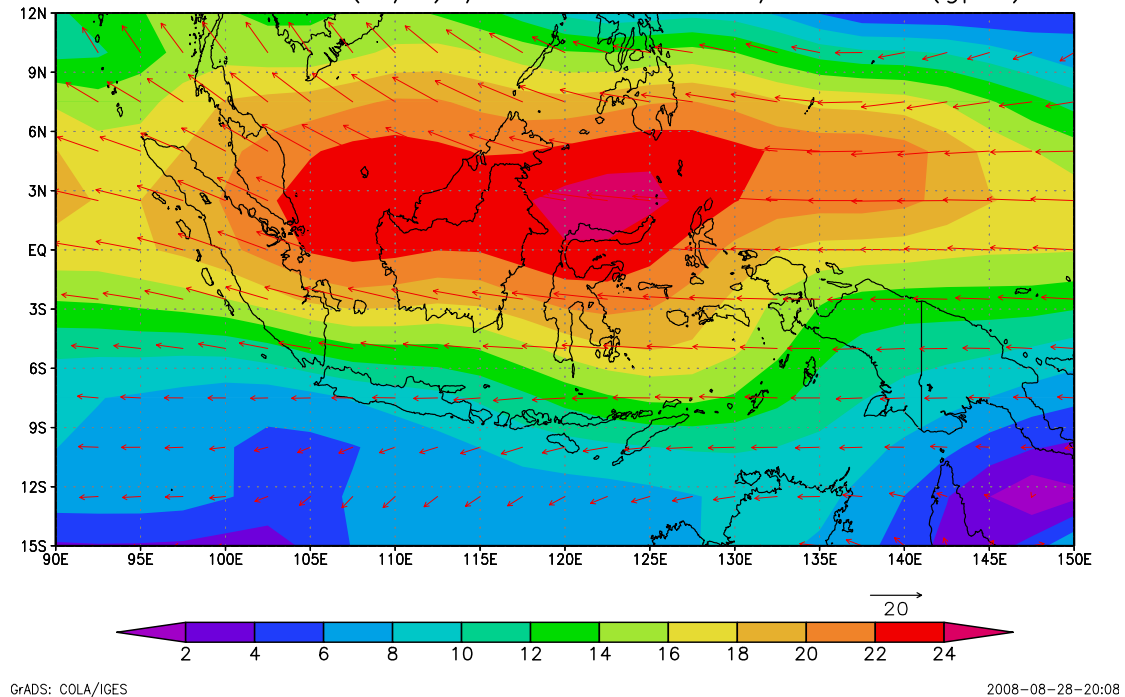
Berikut ini ditampilkan beberapa gambar yang umum dipergunakan di lingkungan meteorologi beserta scriptnya

a. Grafik “250 mb WIND (m/s) / WIND VECTORS / HEIGHT (gpm)”

```
'reinit'  
'open h:/grads/ctl/all.prs'  
'set mpdset hires'  
'set lon 90 150'  
'set lat -15 12'  
'set gxout shaded'  
'define u25=ugrd(lev=250)*2.0'  
'define v25=vgrd(lev=250)*2.0'  
'define wind25=sqrt(u25*u25+v25*v25)'  
'd wind25'  
'run cbarn.gs'  
'set gxout vector'  
'set ccolor 2'  
'd u25;v25'  
'set gxout contour'  
'define h25=hgt(lev=250)'  
'set cint 60'  
'd h25'  
'draw title 250 mb WIND (m/s) / WIND VECTORS / HEIGHT (gpm)'
```

```
'printim h:/grads/image/250mb.gif x800 y600 black'
```

250 mb WIND (m/s) / WIND VECTORS / HEIGHT (gpm)



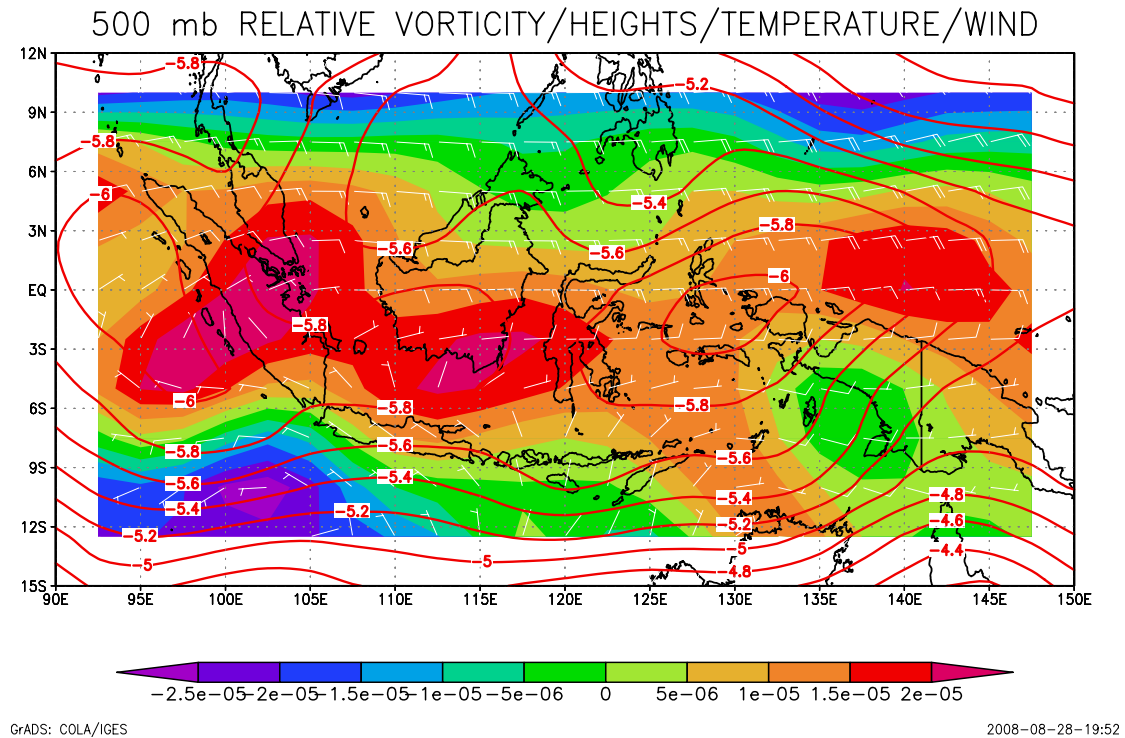
Gambar 5.7

b. Grafik “500 mb RELATIF  
VORTICITY/HEIGHTS/TEMPERATUR/WIND”

```
'reinit'  
'open h:/grads/ctl/all.prs'  
'set mpdset hires'  
'set map 1 1 6'  
'set lon 90 150'  
'set lat -15 12'  
'set gxout shaded'  
'define u5=ugrd(lev=500)*2.0'  
'define v5=vgrd(lev=500)*2.0'  
'd hcurl(u5,v5)  
'run cbarn.gs'  
'set gxout barb'  
'set ccolor 0'  
'd u5;v5'  
'set gxout contour'  
'define h5=hgt(lev=500)  
'set ccolor 1'  
'set cint 60'  
'd h5'  
'set gxout contour'  
'define t5=tmp(lev=500)-273.15'  
'set ccolor 2'
```



```
'set cstyle 1'
'set cthick 8'
'd t5'
'draw title 500 mb RELATIF
VORTICITY/HEIGHTS/TEMPERATUR/WIND'
'printim h:/grads/image/500mb.gif x900 y700 black'
```

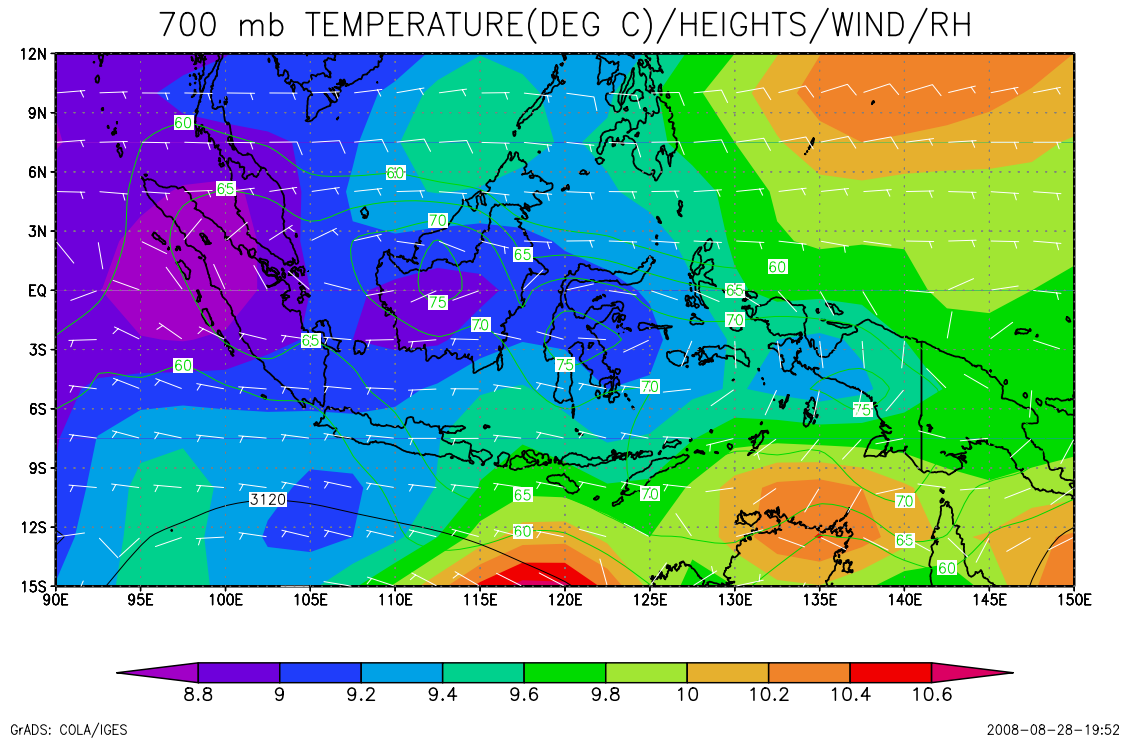


Gambar 5.8

c. Grafik “700 mb TEMPERATUR(DEG C)/HEIGHTS/WIND/RH”

```
'reinit'
'open h:/grads/ctl/all.prs'
'set mpdset hires'
'set map 1 1 6'
'set lon 90 150'
'set lat -15 12'
'set gxout shaded'
'define t7=tmp(lev=700)-273.15'
'd t7'
'run cbarn.gs'
'set gxout contour'
'define rh700=rh(lev=700)'
'set ccolor 3'
'set cmin 60'
'd rh700'
'set gxout contour'
'define h7=hgt(lev=700)'
'set cint 30'
```

```
'set ccolor 1'  
'd h7'  
'set gxout barb'  
'define u7=ugrd(lev=700)'  
'define v7=vgrd(lev=700)'  
'set ccolor 0'  
'd u7;v7'  
'draw title 700 mb TEMPERATUR(DEG C)/HEIGHTS/WIND/RH'  
'printim h:/grads/image/700mb.gif x900 y700 black'
```

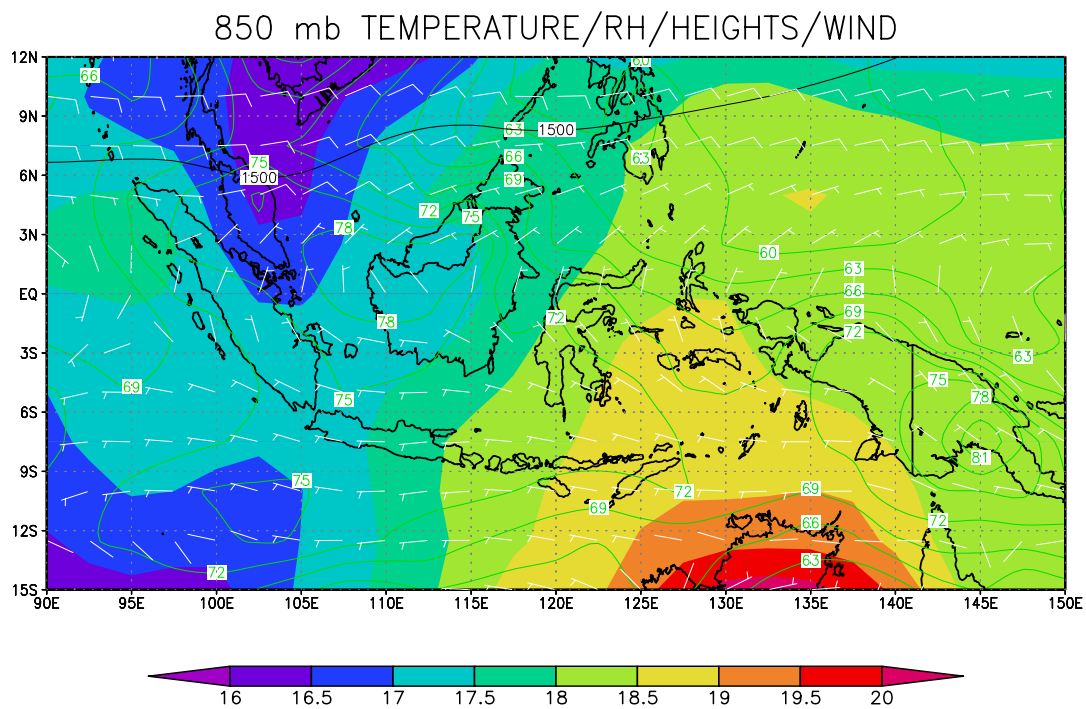


Gambar 5.9

d. Grafik “850 mb TEMPERATUR/RH/HEIGHTS/WIND”

```
'reinit'  
'open h:/grads/ctl/all.prs'  
'set mpdset hires'  
'set map 1 1 6'  
'set lon 90 150'  
'set lat -15 12'  
'set gxout shaded'  
'define t850=tmp(lev=850)'  
'define rh850=rh(lev=850)'  
'd t850-273.15'  
'run cbarn.gs'  
'set gxout contour'  
'set cmin 60'  
'set ccolor 3'  
'd rh850'
```

```
'set gxout contour'  
'define h850=hgt(lev=850)'  
'set cint 30'  
'set ccolor 1'  
'd h850'  
'set gxout barb'  
'define u850=ugrd(lev=850)'  
'define v850=vgrd(lev=850)'  
'set ccolor 0'  
'd u850;v850'  
'draw title 850 mb TEMPERATUR/RH/HEIGHTS/WIND'  
'printim h:/grads/image/850mb.gif x900 y700 black'
```

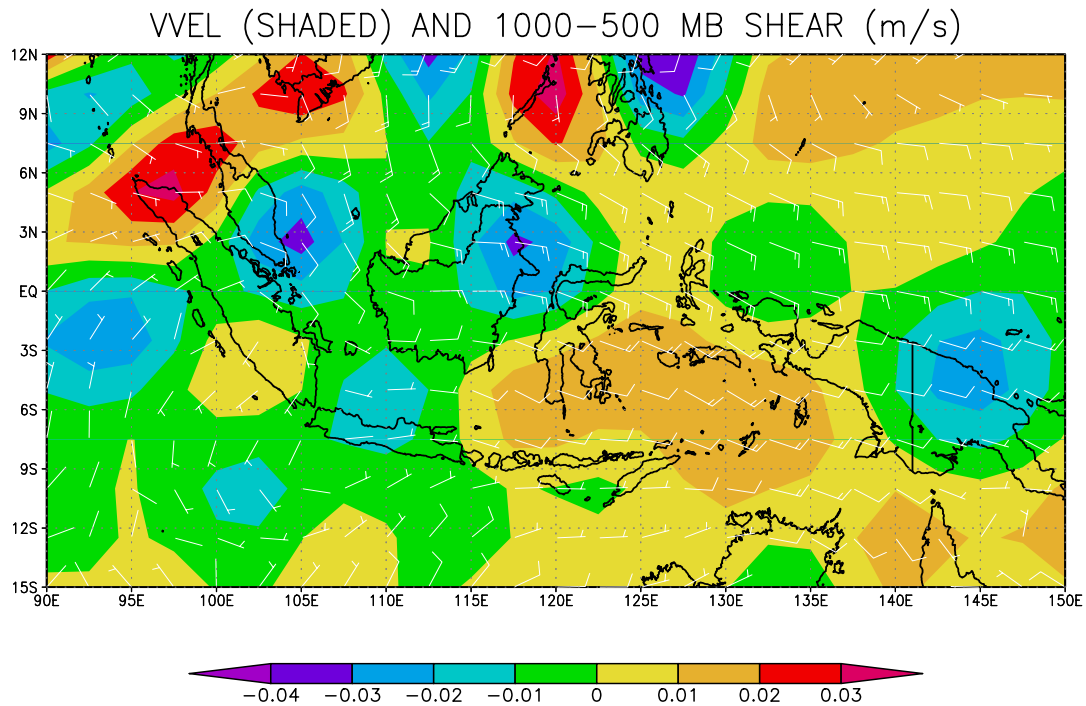


Gambar 5.10

e. Grafik “VVEL (SHADED) AND 1000-500 MB SHEAR (m/s)”

```
'reinit'  
'open h:/grads/ctl/all.prs'  
'set mpdset hires'  
'set map 1 1 6'  
'set lon 90 150'  
'set lat -15 12'  
'set gxout shaded'  
'd vvel'  
'run cbarn.gs'  
'set cmin 20'  
'define u5=ugrd(lev=500)*2.0'  
'define v5=vgrd(lev=500)*2.0'
```

```
'define u1=ugrd(lev=1000)*2.0'  
'define v1=vgrd(lev=1000)*2.0'  
'define ushr=u5-u1'  
'define vshr=v5-v1'  
'set gxout barb'  
'set ccolor 0'  
'set cthick 2'  
'd ushr;vshr'  
'draw title VVEL (SHADED) AND 1000-500 MB SHEAR (m/s)'  
'printim h:/grads/image/vvel_shear.gif x800 y600 black'
```



GrADS: COLA/IGES

2008-08-28-19:55

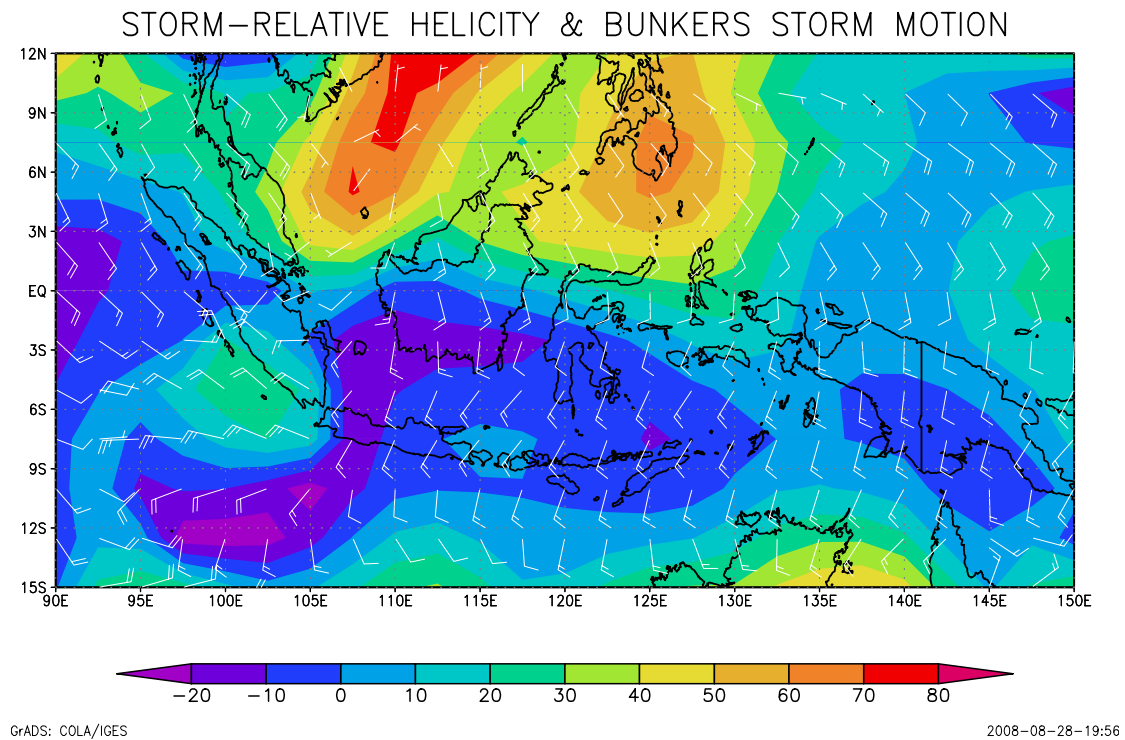
Gambar 5.11

f. Grafik “STORM-RELATIF HELICITY & BUNKERS STORM MOTION”

```
'reinit'  
'open h:/grads/ctl/all.prs'  
'set mpdset hires'  
'set map 1 1 6'  
'set lon 90 150'  
'set lat -15 12'  
# SET WIND LEVELS  
'define u1000=ugrd(lev=1000)'  
'define u925=ugrd(lev=925)'  
'define u925=ugrd(lev=925)'  
'define u850=ugrd(lev=850)'  
'define u850=ugrd(lev=850)'  
'define u700=ugrd(lev=700)'
```

```
'define u700=ugrd(lev=700)'
'define u600=ugrd(lev=600)'
'define u600=ugrd(lev=600)'
'define u500=ugrd(lev=500)'
'define u500=ugrd(lev=500)'
'define u400=ugrd(lev=400)'
'define u400=ugrd(lev=400)'
'define v1000=vgrd(lev=1000)'
'define v950=vgrd(lev=925)'
'define v925=vgrd(lev=925)'
'define v850=vgrd(lev=850)'
'define v850=vgrd(lev=850)'
'define v700=vgrd(lev=700)'
'define v700=vgrd(lev=700)'
'define v600=vgrd(lev=600)'
'define v600=vgrd(lev=600)'
'define v500=vgrd(lev=500)'
'define v500=vgrd(lev=500)'
'define v400=vgrd(lev=400)'
'define v400=vgrd(lev=400)'
#CALCULATE BUNKERS ID METHOD FOR PREDICTING
SUPERCELL MOTION
'define
umean=(u1000+u925+u925+u850+u850+u700+u700+u600+u600+u5
00+u500+u400+u400)/13.0'
'define
vmean=(v1000+v950+v925+v850+v850+v700+v700+v600+v600+v5
00+v500+v400+v400)/13.0'
'define ushear=u500-u1000'
'define vshear=v500-v1000'
'define shear=sqrt(ushear*ushear+vshear*vshear)'
'define umotion=((umean+(7.5/(shear))*vshear))'
'define vmotion=((vmean-(7.5/(shear))*ushear))'
#CALCULATE SRH
'define srh1=((u925-umotion)*(v1000-vmotion)-(u1000-
umotion)*(v950-vmotion))'
'define srh2=((u925-umotion)*(v950-vmotion)-(u925-umotion)*(v925-
vmotion))'
'define srh3=((u850-umotion)*(v925-vmotion)-(u925-umotion)*(v850-
vmotion))'
'define srh4=((u850-umotion)*(v850-vmotion)-(u850-umotion)*(v850-
vmotion))'
'define srh5=((u700-umotion)*(v850-vmotion)-(u850-umotion)*(v700-
vmotion))'
'define srh6=((u700-umotion)*(v700-vmotion)-(u700-umotion)*(v700-
vmotion))'
'define srh3km=srh1+srh2+srh3+srh4+srh5+srh6'
'set gxout shaded'
'd srh3km'
'run cbarn.gs'
```

```
'set gxout barb'  
'define ukt=umotion*2.0'  
'define vkt=vmotion*2.0'  
'set ccolor 0'  
'd ukt;vkt'  
'draw title STORM-RELATIF HELICITY & BUNKERS STORM  
MOTION'  
'printim h:/grads/image/srh.gif x925 y700 black'
```



Gambar 5.12

g. Contoh Script Interaktif “MODEL.GS)

```
'reinit'  
'clear'  
'open h:/grads/ctl/model.ctl'  
'set mpvals -120 -75 25 65'  
'set lon -150 -45'  
'set lat 15 80'  
'set mproj nps'  
'set parea 0 11 0 8.2'  
'set string 1 bc'  
'set strsiz 0.15'  
'set csmooth on'  
'q hardware'  
rec = sublin(result,2)  
_cols = subwrđ(rec,4)  
if (_cols=256)  
'set rgb 30 0 80 0'
```

```

'set rgb 31 0 110 0'
'set rgb 32 0 150 0'
'set rgb 33 0 200 0'
'set rgb 34 0 250 0'
'set rgb 35 150 255 0'
endif
'set mpdset hires'
'set map auto'
'set grid off'

t = 2
hr = 12
while (1)
  'clear'
  'set t t'
  say ' '
  say 'Time = 't' ('hr'hr fcst)'
  say ' '
  say ' '
  say 'A) 500mb Heights and Vorticity'
  say 'B) SLP and Thickness'
  say 'C) Precip and 850mb Temps'
  say 'D) 850mb Streamlines and Isotachs'
  say 'E) 500mb Streamlines and Isotachs'
  say 'F) 850mb RH and Winds'
  if (t>1)
    say 'G) 500mb 12hr Height Change'
    say 'H) 500mb 12hr Vorticity Change'
  endif
  say ' '
  say 'T1) Time = 1'
  say 'T2) Time = 2'
  say ' '
  say 'Q) Quit'
  say ' '
  prompt 'Enter Choice: '
  pull r
  if (r='a' | r='A')
    rc = doa()
  endif
  if (r='b' | r='B')
    rc = dob()
  endif
  if (r='c' | r='C')
    rc = doc()
  endif
  if (r='d' | r='D')
    rc = dod()
  endif
  if (r='e' | r='E')

```

```
rc = doe()
endif
if (r='f' | r='F')
rc = dof()
endif
if (r='g' | r='G')
rc = dog()
endif
if (r='h' | r='H')
rc = doh()
endif
if (r='t1' | r='T1')
t = 1
hr = 0
endif
if (r='t2' | r='T2')
t = 2
hr = 12
endif
if (r='q' | r='Q')
break
endif
endwhile
return
```

\* Draw 500mb Heights and Vorticity

```
function doa()
'clear'
'u5=u(lev=500)'
'v5=v(lev=500)'
'z5=z(lev=500)'
'set map auto'
'set gxout shaded'
'set cint 2'
'set grads off'
'd hcurl(u5,v5)*1e5'
'set gxout contour'
'set cint 2'
'set ccolor 15'
'set grads off'
'd hcurl(u5,v5)*1e5'
'set cthick 3'
'set ccolor 0'
'set cint 3'
'set grads off'
'd z5/10'
'draw string 5.5 8.30 500mb Heights and Vorticity'
pull dummy
```



return

\* Draw SLP and Thickness

```
function dob()
'clear'
'z5=z(lev=500)'
'z10=z(lev=1000)'
'set map auto'
'set gxout contour'
'set grads off'
'd ps'
'set cstyle 5'
'set grads off'
'd (z5-z10)/10'
'set cthick 6'
'set ccolor 1'
'set grads off'
'd (z5-z10)/10'
'draw string 5.5 8.30 SLP and 1000-500mb Thickness'
pull dummy
return;
```

\* Draw Precip and 850 temps

```
function doc()
'clear'
't8=t(lev=850)'
'set map 15 1 1'
'set gxout shaded'
if (_cols=256)
'set clevs 1 2 4 8 16 32'
'set ccols 0 30 31 32 33 34 35'
else
'set clevs 1 2 4 8 16'
'set ccols 0 11 5 13 3 10'
endif
'set grads off'
'd p'
'set gxout contour'
'set ccolor 0'
'set clevs 1 2 4 8 16 32 64'
'set grads off'
'd p'
'set gxout contour'
'set ccolor 0'
'set cthick 12'
'set cint 2'
'set cstyle 1'
'set grads off'
```

```
'd t8-273.16'  
'set ccolor rainbow'  
'set cthick 1'  
'set cint 2'  
'set cstyle 1'  
'set grads off'  
'd t8-273.16'  
'set ccolor 1'  
'set clevs 0'  
'set grads off'  
'set cthick 12'  
'd t8-273.16'  
'set ccolor 0'  
'set clevs 0'  
'set grads off'  
'set cthick 1'  
'd t8-273.16'  
'draw string 5.5 8.30 Precip and 850mb Temperatur'  
pull dummy  
return;
```

\* Do 850mb Streamlines and Isotachs

```
function dod()  
'clear'  
'u8=u(lev=850)'  
'v8=u(lev=850)'  
'set map 1'  
'set gxout shaded'  
'set cint 5'  
'set grads off'  
'd mag(u8,v8)*1.944'  
'set gxout contour'  
'set cint 5'  
'set ccolor 15'  
'set grads off'  
'd mag(u8,v8)*1.944'  
'set gxout stream'  
'set ccolor 0'  
'set grads off'  
'd u8;v8'  
'draw string 5.5 8.30 850mb Isotachs and Streamlines'  
pull dummy  
return
```

\* Do 500mb Streamlines and Isotachs

```
function doe()  
'clear'  
'u5=u(lev=500)'
```

```
'v5=v(lev=500)'  
'set map 1'  
'set gxout shaded'  
'set cint 5'  
'set grads off'  
'd mag(u5,v5)*1.944'  
'set gxout contour'  
'set cint 5'  
'set ccolor 15'  
'set grads off'  
'd mag(u5,v5)*1.944'  
'set gxout stream'  
'set ccolor 0'  
'set grads off'  
'd u5;v5'  
'draw string 5.5 8.30 500mb Isotachs and Streamlines'  
pull dummy  
return
```

\* Do 850mb RH and Winds

```
function dof()  
'clear'  
'rh8=q(lev=850)'  
'u8=u(lev=850)'  
'v8=v(lev=850)'  
'set map 15'  
'set gxout shaded'  
'set grads off'  
'd rh8'  
'set gxout contour'  
'set ccolor 0'  
'set clevs 70 90'  
'set grads off'  
'd rh8'  
'set gxout barb'  
'set digsiz 0.085'  
'set ccolor 0'  
'set cthick 12'  
'set grads off'  
'd u8*1.944;v8*1.944'  
'set ccolor 1'  
'set cthick 1'  
'set grads off'  
'd u8*1.944;v8*1.944'  
'draw string 5.5 8.30 850mb Relatif Humidity and Winds'  
pull dummy  
return
```

\* Do 500mb Height Change

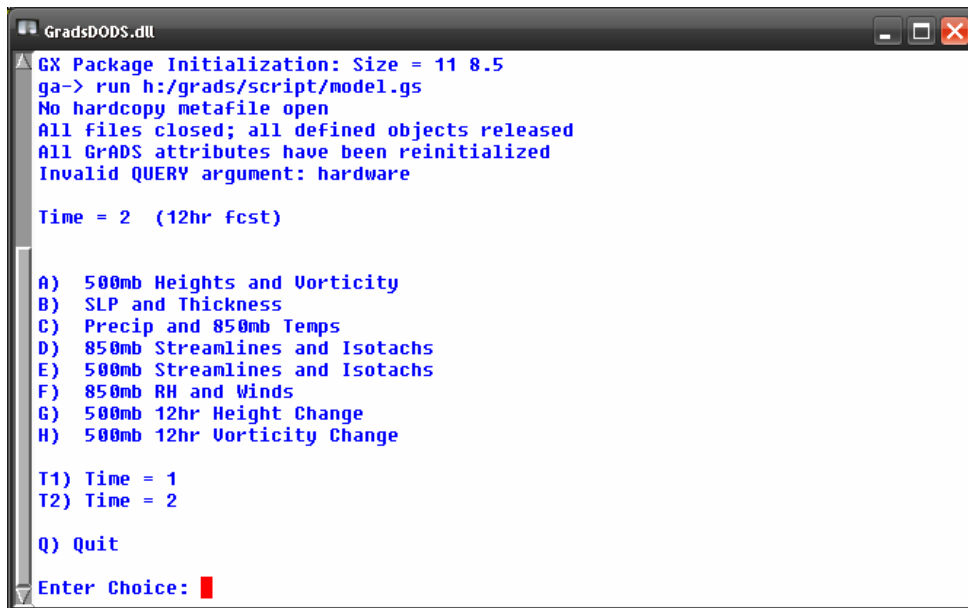
```
function dog()
'clear'
'z5=z(lev=500)'
'set map auto'
'set gxout contour'
'set grads off'
'd z5(t+1)-z5'
'set grads off'
'd z5/10'
'draw string 5.5 8.30 500mb 12Hr Height Change'
pull dummy
return
```

\* Do 500mb Vorticity Change

```
function doh()
'clear'
'set t 2'
'z5=z(lev=500)'
'u5=u(lev=500)'
'v5=v(lev=500)'
'set map auto'
'set gxout contour'
'set grads off'
'd (hcurl(u5,v5)-hcurl(u5(t-1),v5(t-1)))*1e5'
'set grads off'
'd z5/10'
'draw string 5.5 8.30 500mb 12Hr Vorticity Change'
pull dummy
return
```

Apabila script tersebut dijalankan maka akan tampil beberapa pilihan untuk tampilan data seperti pada gambar 5.13 berikut ini.

Program interaktif ini cukup bermanfaat untuk menampilkan gambar sesuai dengan pilihan dengan hanya sekali menjalankan program. Script ini cukup panjang tetapi prinsipnya cukup mudah. Script ini dapat dimodifikasi untuk gambar-gambar yang lain.



```
GradsDODS.dll
GX Package Initialization: Size = 11 8.5
ga-> run h:/grads/script/model.gs
No hardcopy metafile open
All files closed; all defined objects released
All GrADS attributes have been reinitialized
Invalid QUERY argument: hardware

Time = 2 (12hr fcst)

A) 500mb Heights and Vorticity
B) SLP and Thickness
C) Precip and 850mb Temps
D) 850mb Streamlines and Isotachs
E) 500mb Streamlines and Isotachs
F) 850mb RH and Winds
G) 500mb 12hr Height Change
H) 500mb 12hr Vorticity Change

T1) Time = 1
T2) Time = 2

Q) Quit

Enter Choice: █
```

Gambar 5.13

## **Bab VI**

### **PENUTUP**

Setelah mempraktekkan tutorial dan pembuatan script di atas, diharapkan pengguna dapat mengaplikasikan program GrADS ini dengan tampilan yang disesuaikan dengan keperluan masing-masing.

Masih banyak perintah GrADS lain yang dapat dikembangkan lebih lanjut, oleh karena itu dibutuhkan suatu usaha dari pengguna GrADS untuk memaksimalkan perintah-perintah tersebut sehingga dapat berguna terutama untuk tujuan analisis maupun prakiraan.

Banyak sekali aplikasi lainnya yang dapat kita olah melalui program GrADS ini, dan diharapkan di masa mendatang pengguna dapat mengembangkan hal tersebut terutama untuk operasional di kantornya masing-masing.

Untuk saran dan pertanyaan dapat dialamatkan ke alamat :

[erwin\\_makmur@bmg.go.id](mailto:erwin_makmur@bmg.go.id) atau [erwin\\_makmur@yahoo.com](mailto:erwin_makmur@yahoo.com).

Jakarta, Agustus 2008

## **BAHAN BACAAN**

1. ...., 2005, Visualisasi Luaran DARLAM dengan GrADS, Pelatihan GrADS, Puslitbang BMG
2. Brian Doty, 1995, GrADS, University of East Anglia, Norwich, UK
3. GrADS website
4. Kalnay, E., and Coauthors, 1996: The NCEP/NCAR 40-Year Reanalysis Project. *Bull. Amer. Meteor. Soc.*, 77, 437–471.
5. Wesley Ebisuzaki and Coauthors, 2000, NCEP/NCAR Reanalysis CD-ROM

## **Alamat Website GrADS**

1. GrADS homepage  
<http://www.iges.org/grads/>
2. GrADS *Software*  
<http://www.iges.org/grads/downloads.html>
3. GrADS Data Server  
<http://www.iges.org/grads/gds/>

# GrADS reference card version 1.7

(GrADS Version 1.7 beta 7) compiled by Karin Meier-Fleischer, DKRZ (beratung@dkrz.de)

## GrADS program executables

<b>grads</b>	link to one of the following executables
<b>gradsnc</b>	grads with netCDF enabled
<b>gradshdf</b>	grads with HDF enabled
<b>gradsc</b>	grads "classic", without netCDF/HDF/Athena GUI, etc.

## Command line options

<b>Program:</b> grads [-lbpC] [-c 'command']	
<b>-c 'command'</b>	execute 'command' when starting GrADS
<b>-b</b>	run grads in batch mode. No graphics output window is opened.
<b>-l</b>	run grads in landscape mode. The orientation question is not asked.
<b>-p</b>	run grads in portrait mode. The orientation question is not asked.
<b>-C</b>	enable automatic setting of century for years < 100

## General settings

<b>help</b>	gives a summary list of operations
<b>set grads on   off</b>	enable/disable display of the GrADS logo
<b>set display &lt;option&gt; &lt;color&gt;&gt;</b>	
	sets the mode of the display. options are:
grey greyscale	sets the mode to greyscale.
color	shading and contouring is done with rainbow colors. Default.
<black white>	
<b>set frame &lt;option&gt;</b>	
	control the frame on a plot. options are:
on	plots a rectangular frame around clipped region
off	plots no frame
circle	plots a rectangular frame for lat-lon projections, plots a circular frame for a polar plot at the outermost latitude. Whole hemisphere plots only.
<b>set background ic</b>	set background color to color or color index ic
<b>display expression d expression</b>	display data via the graphics output window; the simplest expression is a variable abbreviation
<b>open filename</b>	open descriptor file
<b>sdfopen file.nc &lt;template #timesteps&gt;</b>	
	opens a netCDF or HDF-SDS format file that conforms to the COARDS conventions. The optional arguments are for string a time-series of files together as one GrADS data object.
<b>xdfopen file</b>	opens a non-COARDS-conformant netCDF or HDF-SDS file via a data descriptor file similar to those used with the 'open' command.
<b>close file#</b>	close the last descriptor file opened.
<b>set dfile number</b>	change to descriptor file number for current file
<b>define var=expr var=expr</b>	create new variable, which then can be used in subsequent expressions
<b>undefine var</b>	free the resources used by the defined variable

## modify varname <time type>

	define variable, which is climatological. varname is the defined grid. Time types are:
seasonal	monthly or multi-monthly means
diurnal	over some time period less than a day

<b>query &lt;option&gt; q &lt;option&gt;</b>	query options are:
config	list GrADS configuration information
files	lists open files
file n	gives info on particular file
define	lists currently defined variables
dims	gives current dimension environment
gxinfo	gives graphics environment info
shades	gives colors and levels of shaded contours
pos	waits for mouse click, returns the position

time	gives info about time settings
fwrite	print name of fwrite output file
string s	gives the width of string s
defval v1 i j	gives the value of a defined variable v1 at point i,j
udft	list the user defined function table
lats	state of the GrADS-LATS interface
xy2w v1 v2	XY coords to world coords
xy2gr v1 v2	XY coords to grid coords
w2xy v1 v2	world coords to XY coords
w2gr v1 v2	world coords to grid coords
gr2w v1 v2	grid coords to world coords
gr2xy v1 v2	grid coords to XY coords
ll2xy lon lat	LON/LAT coords to XY coords
pp2xy ppx	page coords to XY coords

<b>set imprun script</b>	automatically executes script before every display command
<b>run file-name &lt;params&gt;</b>	load and run a GrADS script (with parameters)
<b>exec fname &lt;arg0,...arg9&gt;</b>	executes a sequence of GrADS commands from file fname. If a clear command is encountered, GrADS waits until enter is pressed before clearing and continuing with command processing

<b>clear &lt;option&gt; c &lt;option&gt;</b>	c <option> clear the display; option are:
events	flush event buffer
graphics	clear graphic, not widgets
hbuff	clear display buffer, when in double buffer mode

<b>reset &lt;option&gt;</b>	initializes GrADS to its initial state with following exceptions: 1) No files are closed. 2) No defined objects are released. 3) The 'set display' settings are not modified. Options are: events; graphics; hbuff; norset
<b>reinit</b>	same as reset, and in addition closes all open files and releases all defined objects
<b>quit</b>	quit - to leave GrADS
<b>!shell-command</b>	runs a shell command on GrADS command line. The output will not be returned to the script, only displayed.

## Dimension environments

<b>set lon val1 &lt;val2&gt;</b>	sets longitude to vary from val1 to val2
<b>set lat val1 &lt;val2&gt;</b>	sets latitude to vary from val1 to val2
<b>set lev val</b>	sets the level to val - fixed dimension
<b>set t val1 &lt;val2&gt;</b>	sets time to the "val" time in the data set
<b>set x val1 &lt;val2&gt;</b>	set x values or fix it to one value
<b>set y val1 &lt;val2&gt;</b>	set y values or fix it to one value
<b>set z val1 &lt;val2&gt;</b>	set z values or fix it to one value

## Page control

<b>set vpage off</b>	real page is equal to "virtual page"; default state
<b>set vpage xmn xmx ymn ymx</b>	defining one "virtual" page
<b>set parea xmn xmx ymn ymx</b>	control the area within the virtual page

## Graphic types

<b>set gxout graphic-type</b>	where graphic-type could be:
bar	Bar chart
barb	Plot wind barb at station
contour	Contour plot
errbar	Error bar
fgrid	specific value grid fill plot
findstn	Find closest station to x,y point
fwrite	Write data to file instead of displaying
grfill	Filled grid boxes
grid	Grid boxes with values
line	Line graph
linefill	Color fill between two lines
model	Plot station model

scatter	Scatter graph plot
shaded	Shaded contour plot
stat	Display information about data
stream	Streamline plot
tserwx	Plot time series of weather symbols at a point (1-D station)
tserbarb	Plot time series of wind barbs at a point (1-D)
value	Plot station values
vector	Vector wind arrows
wxsym	Plot weather symbols at station

## Default colors, line styles and marker types

<b>colors</b> used by many settings (i.e. ccolor, line, string button, clopts, lfcols, .....):			
0	black	1	white
2	red	3	green
4	blue	5	cyan
6	magenta	7	yellow
8	orange	9	purple
10	yellow/green	11	med.blue
12	dark yellow	13	qua
14	dark purple	15	grey

<b>line styles</b> used by many settings (i.e. cstyle, line, mpt, map, grid, ...):			
0	none	1	solid
2	long dash	3	short dash
4	long short dash	5	dots
6	dot dash	7	dot dot dash

<b>marker types</b> used by many settings (i.e. cmark, mark, .....):			
0	none	1	cross
2	open circle	3	closed circle
4	open square	5	closed square
6	X	7	diamond
8	triangle	10	open circle with vertical line
9	none	11	open oval

## Graphics options

<b>set clip xlo xhi ylo yhi</b>	clipping area for drawing graphics primitives
<b>set ccolor index</b>	sets the contour color to index, see Default colors and line styles. You can also issue: rainbow - rainbow color sequence revrain - reversed rainbow color sequence
<b>set cstyle style</b>	sets the contour or line style, see Default colors and line styles. (gxout = contour, only style 1,2,3 and 5 available).
<b>set cmark marker</b>	sets line marker, see Default colors and line styles .
<b>set cterp on   off</b>	turns spline smoothing on or off
<b>set clab on   off   forced   string   auto</b>	controls contour labeling
<b>set clopts col &lt;thick &lt;size&gt;&gt;</b>	contour line options
<b>set clskip val</b>	skip val contour lines when labelling
<b>set cthick thekns</b>	sets the line thickness for the contours [1-10]
<b>set csmooth on   off   linear</b>	interpolate to a finer grid using cubic or linear interpolation
<b>set cint value</b>	sets the contour interval to the specified "value"
<b>set cmax value</b>	contours not drawn above this value
<b>set cmin value</b>	contours not drawn below this value
<b>set clevs lev1 lev2 ...</b>	sets specified contour levels
<b>set ccols col1 col2 ...</b>	sets specified color for clev levels
<b>set line col &lt;style&gt; &lt;thick&gt;</b>	sets current line attributes. thickness range 1 - 6 (see Default colors and line styles).
<b>set lfcols col1 col2</b>	set color below and above lines (gxout linefill)
<b>set black off   val1 val2</b>	contours not drawn within this interval
<b>set rbcols c1 c2 &lt;c3 ... cn&gt;</b>	specifies a new 'rainbow' color sequence
<b>set rbcols &lt;auto&gt;</b>	built in rainbow sequence is used
<b>set rbrange low high</b>	range of values used to determine which values acquire which rainbow color



<b>set grid on   off</b> <style> <color>	<b>horizontal   vertical</b> draw grid lines using the specified options or not
<b>set bargap val</b>	sets the gap between bars in percent
<b>set barbase value   bottom   top</b>	bar rises from or falls from value
<b>set baropts filled   outline</b>	bar outlined or filled; default: filled
<b>set dignum number</b>	number of digits after the decimal place
<b>set digsize size</b>	size (in inches, or plotter units) of the numbers
<b>set arrlab on off</b>	set arrow labeling on or off
<b>set arrslc size &lt;magnitude&gt;</b>	specifies arrow length scaling
<b>set arrowhead size</b>	specifies arrow head size
<b>set fgvals v1 c1 &lt;v2 c2&gt;...</b>	fgrid output type treats the grid values as rounded integers, and will shade a specified integer valued grid with the specified color.
<b>set zlog on   off</b>	sets log scaling of the Z dimension on or off
<b>set strmden value</b>	specifies the streamline density, where value is from 1 to 10. Default: 5
<b>set stnopts &lt;dig3&gt; &lt;nodig3&gt;</b>	plot the number in the slp location as a three digit number with only the last three digits of the whole number plotted
<b>set mdlopts noblank   blank   dig3   nodig3</b>	plot the number of the model data as a three digit
<b>set stid on   off</b>	controls wether the station id is displayed next to the values or not
<b>set wxcols c1 c2 c3 c4 c5 c6</b>	set colors for weather symbols c1 - c6

### Axis labeling/Annotation/labeling

<b>set xaxis start end &lt;incr&gt;</b>	specifies the axis is to be labeled
<b>set yaxis start end &lt;incr&gt;</b>	specifies the axis is to be labeled
<b>set xlevs lab1 lab2 ...</b>	specifies the label levels to plot for the X axis
<b>set ylevs lab1 lab2 ...</b>	specifies the label levels to plot for the Y axis
<b>set xlint interval</b>	specifies the label interval of the X axis
<b>set ylint interval</b>	specifies the label interval of the Y axis
<b>set xyrev on</b>	reverses the axes on a plot
<b>set xflip on</b>	flips the order of the horizontal axis
<b>set yflip on</b>	flips the order of the vertical axis
<b>set xlab on   off   auto   string</b>	controls and/or draws X axis label
<b>set ylab on   off   auto   string</b>	controls and/or draws Y axis label
<b>set xlabs lab1   lab2   ....</b>	abel the x axis with lab1, lab2, lab3, ....
<b>set ylabs lab1   lab2   ....</b>	label the y axis with lab1, lab2, lab3, ....
<b>draw xlab string</b>	draw x axis label
<b>draw ylab string</b>	draw y axis label
<b>set xlopts col &lt;thick &lt;size&gt;&gt;</b>	controls X axis
<b>set ylopts col &lt;thick &lt;size&gt;&gt;</b>	controls Y axis
<b>set xlpsoffset side</b>	controls position of x axis labels. Where offset - in inches; side - b or t (bottom or top)
<b>set ylpsoffset side</b>	controls position of y axis labels. Where offset - in inches; side - r or l (right or left)
<b>set zlog on   off   swap   undefine</b>	sets log scaling of the Z axis
<b>set annot col &lt;thick&gt;</b>	sets color and line thickness for the above 3 draw commands
<b>set vrange vlo vhi</b>	Set range for plotting 1-D or scatter plots; range of the variable values for y-axis scaling
<b>set vrange2 vlo vhi</b>	Set range for plotting 1-D or scatter plots; range of the variable values for x-axis scaling
<b>set missconn on   off</b>	lines will be connected across missing data
<b>draw title string</b>	draw title at top of graph

### Map projections/drawing

<b>set mproj proj</b>	sets current map projection. Keywords are:
latlon	Lat/lon projection with aspect ratio maintained. Default.
scaled	latlon projection where aspect ratio is not maintained.The plot fills the plotting area.
nps	north polar stereographic
sps	south polar stereographic
robinson	Robinson projection
orthogr	Orthographic projection
mollweide	Mollweide projection
lambert	Lambert conformal conic projection

off	same as scaled, but no map is drawn and lables are not interpreted as lat/lon labels
-----	--

<b>set mpt type off</b> <<col> <style> <thick>>	command to control map background behavior. type is the map type; it can be a number from 0 to 255, or it can be an asterick (*) to indicate this command applies to all the type values. The color can be set to -1, which indicates to GrADS to use the set map settings for this map type, rather than the settings specified by the set mpt command.
<b>set mpvals off   lnmn lnm x ltmn ltm x</b>	sets reference longitudes and latitudes for polar stereogr. plots
<b>set mpdset lowres   mres   hires   nmap</b>	mres and hires have state and country outlines. nmap covers only North America. Default:lowres.
<b>set map auto   color &lt;style &lt;thick&gt;&gt;</b>	draws the map background using the requested line attributes or auto mode
<b>set mpdraw on   off</b>	if off, does not draw the map background
<b>set grid on   off &lt;style &lt;col&gt;&gt;   horizontal   vertical</b>	draw or do not draw lat/lon lines on polar plots using the specified color and linestyle
<b>set poli on   off</b>	selects whether you want political boundries drawn for the mres or hires map data sets. Default is on

### Graphic primitives

<b>draw line x1 y1 x2 y2</b>	draws a line from x1, y1 to x2, y2 using current line drawing attributes
<b>draw rec xlo ylo xhi yhi</b>	draws an unfilled rectangle
<b>draw recf xlo ylo xhi yhi</b>	draws a filled rectangle
<b>draw mark marktype x y size</b>	draws a marker. Marker types (see Default colors and line styles).
<b>draw polyf x1 y1 x2 y2 ... xn yn</b>	draw a filled polyline, where xn=x1 and yn=y1
<b>draw wxsym symbol x y size &lt;color &lt;thickness&gt;&gt;</b>	Draws the specified wx symbol at the specified location

### String primitives

<b>set string col &lt;justification&gt; &lt;thick&gt; &lt;rotation&gt;</b>	sets string drawing attributes. Justification: l - left; c - center; r - right; tl - top left; tc - center top; tr - right top; bl - bottom left; tc - center bottom; tr - right bott. Roation: 90 - counterclockwise, -90 - clockwise..
<b>set strsiz width &lt;height&gt;</b>	sets the string character size
<b>draw string x y string</b>	draws the character string at the x,y position
<b>draw title string</b>	draw a title 'string' on top of the graph

### Color settings

<b>set rgb num red green blue</b>	defines new colors within GrADS, and assigns them to a new color number.color-number num must be a value between 16 and 99 (0 to 15 are predefined)
-----------------------------------	---

### Font settings

<b>set font number</b>	change to font number [0-5]
------------------------	-----------------------------

### Widgets

<b>set button 1 bcol1 bcol2 bcol3 0 fcol1 fcol2 fcol3 thickness</b>	set button colors. 1 - "on" state; 0 - "off" state
<b>draw button number x y width height string</b>	draws a button on position x,y with the attributes
<b>redraw button number 0   1</b>	redraws button number; 1 - "on"; 0 - "off"
<b>set rband wn mode x1 y1 x2 y2</b>	rubber banding. wn = widget #; mode = box or line x1, y1 = lowest point in x/y page units x2, y2 = highest point in x/y page units

<b>draw dropdown number x y width height text</b>	display a dropdown similar to 'draw button' command widget number (0 to 64); x and y are the center location for the 'base' of the dropdown; width and height are the size of the 'base' of the dropdown.
---	---

### Double buffering

<b>set dbuff on off</b>	sets double buffer mode on or off
<b>swap</b>	swaps buffers, when double buffer mode is on

### Animation

<b>set looping on off</b>	control animation; set animation on or off
<b>set loopdim x y z t</b>	animate through x,y,z or t; default: t
<b>set loopincr incr</b>	set looping increment

### Hardcopy output

<b>enable print fname</b>	enables the print command to the given file fname
<b>print</b>	copy the contents of current display into a file in a metacode format
<b>disable print</b>	close print output file
<b>outxwd file</b>	output the graphicw window to a file in the X windows dump format
<b>wi file.format</b>	output to a file with format (using ImageMagick), e.g. wi test.gif

### Create/Write a grid file

<b>set fwrite fname</b>	output grid fname; if not set, fname=grads.fwrite
<b>set gxout fwrite</b>	enables grid file output
<b>disable fwrite</b>	close output grid file

### Mathematical Functions

<b>abs(expr)</b>	absolute value of result of expr. Operates on gridded and station data
<b>acos(expr)</b>	applies the cos <sup>-1</sup> function to the result of expr
<b>asin(expr)</b>	applies the sin <sup>-1</sup> function to the result of expr
<b>atan2(expr1,expr2)</b>	applies the tan <sup>-1</sup> function to the result of the two expr, using tanθ = y/x
<b>cos(expr)</b>	takes the cosine of the expr
<b>exp(expr)</b>	performs the ex operation, where expr is x. gridded and station data
<b>gint(expr)</b>	general integral. same as ave except do not devide by the total area
<b>log(expr)</b>	takes the natural logarithm of expr
<b>log10(expr)</b>	takes the logarithm base 10 of the expr
<b>pow(expr1,expr2)</b>	raises the values of expr1 to the power of expr2
<b>sin(expr)</b>	takes the sine of the provided expr (in radians)
<b>sqrt(expr)</b>	takes the square root of the result of the expr
<b>tan(expr)</b>	takes the trigonometric tangent of the expr

### Averaging Functions

<b>ave(expr,dexpr1,dexpr2&lt;,finc&lt;,flags&gt;&gt;)</b>	generalized averaging function. expr is averaged through the dimension range specified by dim1 and dim2
<b>aave(expr,xdim1,xdim2,ydim1,ydim2)</b>	does area average. xdim1 and xdim2 must be for lon or x, ydim1 and ydim2 must be for lat or y (e.g. aave(t,lon=0,lon=180,lat=0,lat=90))
<b>mean(expr,dexpr1,dexpr2&lt;,finc&lt;,flags&gt;&gt;)</b>	same as ave, except that area weighting is disabled
<b>amean(expr,xdim1,xdim2,ydim1,ydim2)</b>	same as aave, except that area weighting is disabled
<b>vint(psexpr,expr,top)</b>	performs a mass-weighted vertical integral in mb pressure coordinates, where: expr-expression for quantity to be integrated psexpr expression yielding the surface pressure, in mb,which will be used to bound the integration on the bottom topconstant, giving the bounding top pressure, in mb. This cannot be provided as an expression

### Grid Functions

<b>const(expr,const&lt;,flag&gt;)</b>	function allows you to set various parts of a grid to a constant
<b>maskout(expr,mask)</b>	whenever the mask values are less than zero, the values in expr are set to the missing data value
<b>skip(expr,skipx,skipy)</b>	sets alternating values of the expr to the missing data value.This function is used while displaying wind arrows or barbs to thin the number of arrows or barbs

## Filtering Functions

<b>smth9(expr)</b>	performs a 9 point smoothing to the gridded result of expr
--------------------	--

## Finite Difference Functions

<b>cdiff(expr,dim)</b>	performs a centered difference operation on expr in the direction specified by dim
------------------------	--

## Meteorological Functions

<b>tvrh2q(tvexpr,rhexpr)</b>	given virtual temperature and relative humidity, tvrh2q returns specific humidity, q, in <u>g/g</u>
<b>tvrh2t(tvexpr,rhexpr)</b>	given virtual temperature and relative humidity, tvrh2t returns the temperature in degrees Kelvin

## Special Purpose Functions

<b>tfoot(expr)</b>	when time is varying dimension in the dimension environment, tfoot function evaluates the expr at fixed times, then constructs the time series to obtain a final result that is the time varying
--------------------	--

## Vector Functions

<b>hcurl(uexpr,vexpr)</b>	calculates the vertical component of the curl (i.e.vorticity) at each grid point using finite differencing on the grids provided
<b>hdivg(expr1,expr2)</b>	calculates the horizontal divergence using the finite differencing
<b>mag(uexpr,vexpr)</b>	performs the calculation: $\sqrt{uexpr^2+vexpr^2}$

## Station Data Functions

<b>gr2stn(grid_expr,stn_expr)</b>	performs an interpolation from grid space back to station locations
<b>oacres(grid_expr,stn_expr&lt;,&gt;radii&lt;first guess&gt;&gt;)</b>	a Cressman objective analysis is performed on the station data to yield a gridded result representing the station data
<b>stnave(expr,dexpr1,dexpr2&lt;,&gt;m cnt&gt;)</b>	takes an average of station data over time
<b>stnmin(expr,dexpr1,dexpr2&lt;,&gt;m cnt&gt;)</b>	examines a time series of station data and returns the minimum value encountered for each station
<b>stnmax(expr,dexpr1,dexpr2&lt;,&gt;m cnt&gt;)</b>	examines a time series of station data and returns the maximum value encountered for each station

## Create PostScript files

<b>Program: gxls [-c] [-r] [-d] [-i mfile] [-o ofile]</b>
converts the GrADS meta file into a PostScript file. Command line options:
-c color on a white background (=old gxpwcw)
-r color on a black background (=old gxpsc)
-d add ctrl-d to the end of the file, useful if printing on HP 1200C/PS printer
-i mfile where mfile is the name of the input GrADS meta file
-o ofile where ofile is the name of the output PostScript file

<b>Program: gxeps [-l] [-2] [-a] [-l] [-c] [-r] [-d] [-L] [-n] [-s] [-v] [-i mfile] [-o ofile]</b>
converts the GrADS meta file into a PostScript file. Command line options:
-1 PostScript Level 1 output
-2 PostScript Level 2 output
-a DIN A4 paper size
-c color on a white background
-d add ctrl-d to the end of the file, useful if printing on HP 1200C/PS printer
-l US letter paper size
-L ask for a label to be printed on the plot
-n ask for a note to include in postscript file header
-r color on a black background
-s add a file & time stamp
-v verbose
-i mfile where mfile is the name of the input GrADS meta file
-o ofile where ofile is the name of the output PostScript file

## Create GIF files

<b>Program: gxgif [-i mfile] [-o ofile]</b>
converts the GrADS meta file into a GIF file. Command line options:
-i mfile where mfile is the name of the input GrADS meta file
-o ofile where ofile is the name of the output GIF file

## Variables

<b>complete specification for a variable name</b>
abbrev.file#(dimexpr,dimexpr,...)
abbrev is the abbreviation for the variable as specified in the data descriptor file file# is the file number that contains this variable. The default initially is 1. dimexpr is a dimension expression that locally modifies the current dimension environment.

## General Information

The GrADS scripting language, used via the GrADS run command, provides a similar capability to the exec command, except that a script may have variables, flow control and access GrADS command output. Scripts may be written to perform a variety of functions, such as allowing a user to point and click on the screen to select something, to animate and desired quantities, to annotate plots with information obtained from GrADS query commands.

**Important:** GrADS needs a carriage return after the last command line in the script file, otherwise GrADS won't execute this command line.

## Variables

Script language variable names are 1 to 8 characters, beginning with an alphabetic character and containing letters or numbers only. The name is case sensitive. The contents of a script variable is always a character string! For some operations, the character string will be interpreted as a number.

**Predefined variables**

<b>lat</b>
<b>lon</b>
<b>lev</b>
<b>result</b>
<b>rec</b>

**String variables** or **string constants** are enclosed either with single or double quotes.

**name** = 'Peter Pan'  
 or **name** = "Peter Pan"

**Compound variables** can be used to construct arrays in scripts. A compound variable has a variable name with segments separated by periods.

**varname.i.j**

Example: i = 10  
 j = 3  
 varname.i.j = 343  
 or varname.10.3 = 343

**Note:** The compound variable name MAY NOT be longer than 16 characters either BEFORE or AFTER substitution. GrADS scripting language is not particular efficient in handling large numbers of variables. Thus compound variables should not be used to create large arrays!

**Global variables** start with an underscore ( \_ ) and will keep its value throughout an entire script file using (also in functions).

**\_varname**

Example: **\_var1** = 1024

**Note:** The global variables cannot be used in function headers  
 "function myfunc (**\_var1**)" would be invalid!  
 It wouldn't make sense, cause it's a global variable!!!

## Assignment

The format to assign a record is: **variable = expression**

The expression is evaluated, and the result is assigned to be the value of the indicated variable.

## Logical values

Logical values are

**TRUE 1**  
**FALSE 0**

## Operators

The following operators are implemented:

	logical OR	&	logical AND
!	unary NOT	-	unary minus
%	concatenation	=	equality
!=	not equal	>	greater than
>=	greater than or equal	<	less than
<=	less than or equal	+	addition
-	subtraction	*	multiplication
/	division		

Arithmetic operations are done in floating point. If the result is integral, the result string will be integer. A logical operator will give a character 0 (zero), if the result is FALSE, and a character 1 (one), if the result is TRUE.

## Expressions

Script expression consists of operands, operators and parentheses. The precedence of the operators is

```
- !(unary)
/ *
+ -
%
= != > >= < <=
&
|
```

Within the same precedence level, operations are performed left to right. Parentheses modify the order of operation in the expected ways.

To concatenate two or more strings using the concatenate operator (%) or just two single quotes ( ' ' ) instead of the operator.

Example: col1 = '16 17 18 19 20 '  
 col2 = '21 22 23 24 25 '  
 col3 = '26 27 28 29 30 '  
 colors = col1%col2%col3  
 or colors = col1"col2"col3

'set ccols 'colors  
 is equal to 'set ccols 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30'

## Standard input/output

To write information to the terminal (standard output):

**say expression**

To write an input request string:

**prompt expression**

The 'prompt' command works the same way as 'say' except it does **not append** a carriage return!

To read an input string/value from the standard input:

**pull variable**

The script pauses for the user keyboard input (up to the carriage return), and the string entered by the user is assigned to the indicated variable name.

Examples: line = 'Peter Pan, the flying one'  
 say line

prompt 'Enter latitude: '  
 pull lat  
 prompt 'Enter longitude: '  
 pull lon  
 'set lat 'lat  
 'set lon 'lon

To combine variables and comments writing to standard out:

say 'She said, it is 'line  
 produces She said, it is Peter Pan, the flying one

## Control flow

**IF Block:**

**if** expression command .....  
**else** command .....  
**endif**

each must be separated lines  
 optional  
 required!

**Note:** There is NO 'else if' element implemented in GrADS!

Example: if(i=10); j=20; endif

is equal to if(i=10)  
 j=20  
 endif

**WHILE Loop:**

**while** expression command .....  
**endwhile**

To continue the while loop use the **continue** command; to exit the while loop use the **break** command

Example: t=1  
 while(t<10)  
 'set t 't  
 'd z'  
 t = t + 1  
 endwhile

## Functions

Functions are invoked as a script expression is being evaluated. Functions always have a single string result, but may have one or more string arguments! Functions are invoked by:

**name(arg1, arg2, arg3, ..., argn)**

If the function has no arguments, you must still provide the parentheses:

**name()**

To define a user own function by using the function definition record:

**function name(var1, var2, var3, ..., varn)**

To return from a function, use the return command:

**return(expression)**

The expression is optional, if not provided, a NULL string will be returned.

Example: x = 10  
 y = 30  
 z = addit(x,y)  
 say 'Result of addition: z = 'z  
 ....  
 function addit(var1,var2)  
 sum=var1+var2  
 return (sum)

terminal output Result of addition: z = 40

## Sending commands

The statement record consists only of an expression

### expression

The expression is evaluated, and the resulting string is submitted to GrADS as a command. After this record is executed, the script variable **'result'** is given the value.

Examples:       hallo = 'draw string 4.0 8.0 HALLO'  
                  hallo

or   'query'  
      say result

produces

```
GrADS Version 1.7Beta6 ---
LATS=GRIB_NCSA_HDF__SDF_READ=NCSA_netCDF_HDF
query or q Options:
q conf1g  List configuration of this build
q files:  Lists open files
q file n:  Gives info on particular file
q define:  Lists currently defined variables
q fwrite:  Fwrite Status
q lats:    State of the GrADS-LATS Interface
q dims:    Gives current dimension environment
q time:    Gives current time
q gxinfo:  Gives graphics environment info
q shades:  Gives colors and levels of shaded contours
q pos:     Waits for mouse click, returns position
q w2gr:    Convert world to grid coordinates
q gx2w:    Convert grid to world coordinates
q w2xy:    Convert world to x,y screen coordinates
q xy2w:    Convert x,y screen coordinates to world
           coordinates
q gr2xy:   Convert grid to x,y screen coordinates
q xy2gr:   Convert x,y screen to grid coordinates
q pp2xy:   Convert pre-projected grid coordinates to
           screen x,y coordinates
q defval:  Gives defined grid value given grid i,j
```

## Intrinsic functions

To get a single word from a string:

**res = subwrđ(string,word)**

The result is the nth 'word' from the string. If string is too short, the result is the NULL string. 'word' must be an integer value.

To get a single line from a string containing several lines:

**res = sublin(string,line)**

The result is the nth 'line' from the string. If the string has too few lines, the NULL string is returned. 'line' must be an integer value.

To get a part of a string:

**res = substr(string,start,length)**

The substring of 'string' starting at location 'start' for length 'length' will be returned. If the string is too short, the result will be short or the NULL string. 'start' and 'length' must be an integer value.

Examples:       'query time'  
                  res = subwrđ(result,3)  
                  year = substr(res,9,4)  
                  say year

produce e.g. 1880

The function sublin is very usefull, if you want to control opening, reading, writing and closing an extern ASCII file.

For example, the first record in the ASCII file 'the\_title.txt' to be read is

Szenario A 1880 - 2099

The following part of a script will open, read and close the file, controlling the status of each statement:

```
ret = read('the_title.txt')
code = sublin(ret,1)
if(code != 0)
  say 'read error #'code
  'quit'
endif
title = sublin(ret,2)
'draw title 'title
ret = close('the_title.txt')
code = sublin(ret,1)
if(code != 0)
  say 'close error #'code
  'quit'
endif
```

## I/O functions

To read records from an ASCII file:

**res = read(filename)**

The result is a string containing two lines. The first line is the return status and the second line is the record. The record may have a maximum of 80 characters. Use the sublin intrinsic function to separate the lines.

The return status of read:

```
0  ok
1  open error
2  end of file
8  file open for write
9  I/O error
```

To write records to an ASCII output file:

**res = write(filename, record <,append>)**

The record is written to the file 'filename'. On the first call to write for a particular file, the file is opened in write mode; this will destroy an existing file 'filename'! If you use the optional append flag, the file will be opened in append mode, and all writes will be appended to the end of the file.

The return status of write:

```
0  ok
1  open error
8  file open for read
```

To close an opened file:

**res = close(filename)**

The close command closes the named ASCII file and can also be used to rewind the file.

The return status of close:

```
0  ok
1  file not open
```

Examples:       'q file 1'  
                  ret = result  
                  res = write('file\_1.txt', ret)  
                  status = sublin(res,1)  
                  if(status != 0)  
                    say 'write error #'status  
                    'quit'  
                  endif  
                  res = close('file\_1.txt')  
                  status = sublin(res,1)  
                  if(status != 0)  
                    say 'close error #'status  
                    'quit'  
                  endif

## Example script

The following example script draws 1200 shaded contour frames (1200 time records). The year, which will be used in the title string, is read from the 'query time' result. The private colors are defined in the function palette(). The 'set clip ..' command is used with the 'set dbuff on' and 'swap' commands to restrict the redraw of the plot to areas with changes from frame to frame.

**At the DKRZ - Hamburg, videos were recorded using this kind of animation within GrADS. To achieve smooth animations, the single frame technique had been applied.**

```
'reinit'
'open descriptor.ctl'
count = 0
rec = 1200
incr = 1; t = 1
palette()
'set vpage 0.0 11.0 0.0 8.5'
'set parea 1.0 10.0 1.4 7.9'
'set dbuff on'
'set mpdset lowres'
'set map 0 1 10'
'set lat -90 90'
'set lon -180 180'
'set mpvals -180 180 -90 90'
'set mproj robinson'
'set grid on 5 0'
while (count < rec)
  'set t 't
  'q time'
  res = subwrđ(result,3)
  year = substr(res,9,4)
  'set grads off'
  'set string 1 c 8'
  'set strsiz 0.23 0.26'
  'draw string 5.5 7.6 Aerosol - Control 'year
  'set gxout shaded'
  'set cint 1.0'
  'set cmin -4.0'
  'set cmax 4.0'
  'set clevs -4.0 -3.0 -2.0 -1.0 0.0 1.0 2.0 3.0 4.0'
  'set ccols 17 18 19 21 22 23 24 25 26 27'
  'display data'
  'set gxout contour'
  'set cterp off'
  'set csmooth off'
  'set cint 1.0'
  'set clab off'
  'display data'
  'run cbar.gs'
  'set clip 1.0 10.0 1.4 7.9'
  'swap'
  count = count + incr
  t = t + incr
endifwhile

function palette()
'set rgb 16 0 0 20'
'set rgb 17 0 29 85'
'set rgb 18 0 44 128'
'set rgb 19 0 83 230'
'set rgb 21 0 151 250'
'set rgb 22 104 173 255'
'set rgb 23 177 213 255'
'set rgb 24 255 250 110'
'set rgb 25 255 209 116'
'set rgb 26 255 160 80'
'set rgb 27 255 100 65'

return
```