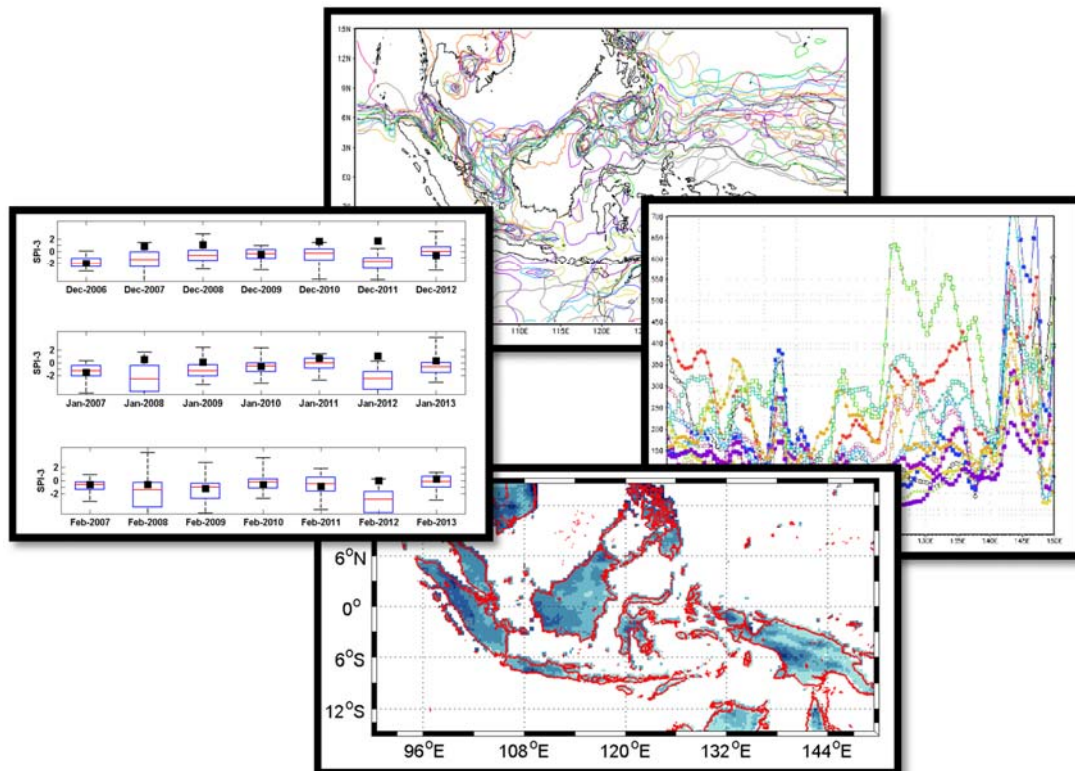


MODUL

Pelatihan Statistical Downscaling

WEATHER AND CLIMATE PREDICTION LABORATORY
<http://weather.meteo.itb.ac.id/>



Continuing Education Program
Lembaga Penelitian dan Pengabdian kepada Masyarakat
Institut Teknologi Bandung
2017

DAFTAR ISI

DAFTAR ISI	I-i
BAGIAN I	I-1
BAB 1 SISTEM PREDIKSI ENSEMBLE	I-1
1.1 Model Prediksi Iklim	I-1
1.2 Model Global (<i>Climate Forecast System (CFS)</i>)	I-2
1.3 Sistem Prediksi Cuaca	I-2
1.4 Chaos, Prediktabilitas Atmosfer, dan Prediksi Ensemble	I-3
BAB 2 SD (<i>Statistical Downscaling</i>)	I-7
2.1 Pengenalan SD (<i>Statistical Downscaling</i>)	I-7
2.2 Kerangka Klasifikasi Untuk Metode SD	I-7
BAB 3 CONSTRUCTED ANALOGUE DAN STATISTICAL BIAS CORRECTION	I-11
3.1 <i>Statistical Bias Correction</i>	I-11
3.2 Constructed Analogue	I-12
BAGIAN II	II-1
BAB 4 PENGENALAN MATLAB	II-1
4.1 Lingkungan Kerja Matlab	II-1
4.2 Pemahaman Tentang Array dan Matrik	II-6
4.3 Fungsi Dasar	II-15
4.4 Input/Output Dasar	II-17
4.5 Visualisasi Matlab	II-18
BAB 5 DASAR-DASAR SCRIPTING PADA MATLAB	II-23
5.1 Kontrol Program	II-23
5.2 Fungsi M-File	II-26
BAB 6 TEKNIK SD (<i>STATISTICAL DOWNSCALLING</i>)	II-28
6.1 Statistical Bias Correction dan Constructed Analogue	II-30

BAGIAN I

BAB 1

BAB 1 SISTEM PREDIKSI *ENSEMBLE*

Di kalangan masyarakat, Meteorologi dapat dikatakan hampir identik dengan prakiraan cuaca meskipun teknologi prediksi sebenarnya bukan monopoli ahli Meteorologi karena mengetahui keadaan masa depan adalah salah satu hal yang sangat diinginkan oleh manusia untuk berbagai tujuan. Contohnya saja, para ekonom memprediksi laju pertumbuhan ekonomi untuk satu tahun ke depan, pialang pasar modal memprediksi nilai valas dan saham untuk satu hari ke depan, para pengamat politik memprediksi perolehan suara partai peserta pemilu, dan sebagainya. Dalam hal ini, metoda empirik (regresi) yang dikembangkan untuk prediksi ekonomi, nilai valas, hasil Pemilu, temperatur di Bandung, mungkin saja sama karena dasar dari teknik prediksi menggunakan metoda empirik adalah pengenalan pola. Berbeda dengan itu, metoda *prediksi cuaca numerik* (PCN) atau *numerical weather prediction* (NWP) dibangun berdasarkan kaidah-kaidah fisis yang mengatur gerak atmosfer dan berbagai proses yang terkait di dalamnya. Kaidah-kaidah fisis ini kemudian diterjemahkan menjadi sistem persamaan matematis yang dapat diselesaikan secara numerik (dengan bantuan komputer). Sampai dua dekade ke belakang, PCN sering juga disebut metoda prediksi *deterministik*.

Dewasa ini, prediksi cuaca numerik secara *de facto* sudah menjadi teknologi prediksi cuaca standar di dunia. Ide pengembangan metoda prediksi cuaca numerik awalnya dimajukan oleh Vilhelm Bjerknes, seorang fisikawan asal Norwegia, kemudian diimplementasikan oleh ilmuwan Inggris Lewis Fry Richardson sekitar tahun 1920-a, meskipun tanpa komputer. Eksperimen PCN pertama dilakukan oleh John von Neuman menggunakan komputer digital generasi pertama yang disebut ENIAC di sekitar tahun 1950-an (Nebeker, 1995). Kesuksesan eksperimen von Neuman menginspirasi banyak ilmuwan yang lebih muda seperti Jule Charney dan Carl Gustav Rossby pada waktu itu untuk melahirkan era baru dalam Meteorologi modern.

1.1 Model Prediksi Iklim

Pusat prediksi cuaca dan iklim seperti *National Centers for Environmental Prediction* (NCEP) dan *European Center for Medium-Range Weather Forecasts* (ECMWF) membuat prediksi iklim secara dinamik menggunakan *general circulation models* (GCM) kopel atmosfer-laut. Kedua model ini telah menunjukkan kemampuan (*skill*) dalam memprediksi suhu permukaan laut (*sea surface temperature* (SST)) pada skala global maupun kontinental (Troccoli dkk, 2008). Meski begitu, mereka tidak dapat menyediakan prediksi yang cukup akurat untuk kondisi iklim lokal karena resolusinya terlalu kasar (mencapai beberapa ratus kilometer) (Juneng dkk, 2010). Padahal, prediksi

tersebut menyediakan informasi penting untuk berbagai sektor yang telah disebutkan di atas (Lavers dkk, 2009).

Terdapat beberapa cara untuk meningkatkan kemampuan prediksi dari GCM. Dua di antaranya adalah *dynamical downscaling* dan *statistical downscaling*. *Dynamical downscaling* melibatkan model iklim regional beresolusi tinggi yang menggunakan luaran GCM sebagai syarat batas (Gao dkk, 2008). *Statistical downscaling* bertujuan untuk menentukan hubungan statistik antara kondisi iklim lokal (yang bertindak sebagai prediktan) dengan variabel atmosfer skala besar (sebagai prediktor) yang diprediksi secara akurat oleh model dinamik (Feddersen dan Anderson, 2005). Pendekatan ini lebih sering digunakan karena lebih mudah dan lebih rendah biaya komputasinya daripada *dynamical downscaling* (Chen dkk, 2012).

1.2 Model Global (*Climate Forecast System (CFS)*)

Climate Forecast System (CFS) adalah suatu sistem operasional untuk prediksi iklim yang dikeluarkan oleh *National Centers for Environmental Prediction (NCEP)*. CFSv2 dibuat operasional oleh NCEP pada Maret 2011. Versi ini telah meng-*upgrade* seluruh aspek dari asimilasi data dan komponen model prediksi dari sistem. Data *Re-analysis* telah dibuat dengan periode 32 tahun (1979-2011), yang menyediakan initial kondisi untuk melakukan *reforecast* data untuk periode 29 tahun (1982–2011). Data 9 bulan *hindcasts* menggunakan data inisiasi 5 harian dan *di-running 4 cycle* perhari dimulai dari tanggal 1 Januari setiap tahun selama 29 tahun periode dari 1982 – 2011. Data-data ini digunakan untuk memperoleh kalibrasi yang konsisten dan stabil serta mengestimasi *skill* dari prediksi operasional seasonal dan sub-seasonal model. Sebagai tambahan, prediksi 1 musim (123 hari) *hindcast* yang *di-running*, menggunakan inisial kondisi dari setiap *cycle 00 UTC* antara 5 harian ini selama periode 12 tahun (1999 – 2010). Gambar Data ini digunakan untuk mengkalibrasi prediksi CPC musim pertama secara operasional untuk prediksi hidrologi (presipitasi, evaporasi, runoff, streamflow, dsb). Sebagai tambahan, 3 *cycle* prediksi *hindcasts* 45 hari ke-depan *di-running* setiap 06, 12, dan 18 UTC *cycles*, selama 12 tahun periode (1999 – 2010) untuk kalibrasi prediksi sirkulasi tropis (MJO, PNA, dsb). Adapun untuk konfigurasi data operasional (*real-time*) adalah terdapat 1 *members running* data CFS prediksi 9 bulan ke-depan tiap 00, 06, 12, dan 18 *cycles*, terdapat 3 member data prediksi 1 musim pada *cycle 00*, dan terdapat 3 *members* data prediksi 45 hari ke-depan pada 06, 12, dan 18 *cycles*. Saha dkk (2006) menyebutkan bahwa CFS merupakan sistem yang telah *coupled* terhadap prediksi musiman yang dinamis antara daratan, lautan, dan atmosfer. Berdasarkan penelitian (Jiang, 2012), pola presipitasi dan sirkulasi dapat diprediksi beberapa bulan sebelumnya, CFSv2 memiliki *skill* yang lebih baik dalam memprediksi Monsun Asia Tenggara.

1.3 Sistem Prediksi Cuaca

General circulation model atau *global circulation model (GCM)* adalah representasi numerik dari banyak persamaan dasar yang menggambarkan perilaku sistem iklim dan interaksi antar komponen-komponennya (atmosfer, hidrosfer, kriosfer, biosfer, dan litosfer) (McGuffie dan Henderson-Sellers, 2005, seperti dikutip dari Gunawan dan Linarka, 2011). GCM dapat

menghasilkan prediksi sirkulasi atmosfer rata-rata dengan *skill* yang tinggi, terutama di daerah tropis (Stockdale dkk, 1998). Namun, GCM memiliki *skill* yang rendah dalam mensimulasikan variabel iklim lokal, seperti yang telah dibuktikan pada penelitian oleh Juneng dkk (2010). Salah satu alasan utama yang menjelaskan kegagalan GCM pada skala regional maupun lokal adalah resolusi spasialnya yang terlalu kasar (100 – 600 km), sehingga variasi topografi dan proses-proses subgrid tidak dapat tergambarkan dengan baik (Zorita dan von Storch, 1999). Salah satu cara untuk menjembatani celah antara prediksi variabel iklim skala global dengan skala regional maupun lokal adalah dengan *statistical downscaling*.

Climate Forecast System (CFS) merupakan sistem prediksi musim *ensemble* yang dikembangkan oleh *National Centers for Environmental Prediction* (NCEP). Ia merupakan salah satu bentuk GCM dengan sistem prediksi kopel atmosfer-laut-darat (Saha dkk, 2006). CFS merupakan sumber informasi penting untuk prediksi iklim musiman di wilayah Asia yang dipengaruhi oleh monsun (Jiang dkk, 2013). Versi terbaru CFS adalah *Climate Forecast System version 2* (CFSv2) yang mulai operasional sejak Maret 2011. CFSv2 telah mendapat *upgrade* di hampir semua aspek asimilasi data maupun komponen model prediksi (Saha dkk, 2014).

Prediksi *ensemble* adalah prediksi cuaca numerik yang menghasilkan banyak kemungkinan di masa depan dengan menggunakan beberapa sampel untuk mengestimasi ketidakpastian hasil prediksi (WMO, 2012). Salah satu cara untuk membangun prediksi *ensemble* adalah *time lag ensemble*. Metode *time lag ensemble* menggunakan prediksi dari waktu awal yang berbeda-beda (*lagged*) sebagai member *ensemble*-nya (Callado dkk, 2013). Metode ini mempunyai dua faktor berlawanan yang dapat mempengaruhi kemampuan prediksinya. Di satu sisi, penggunaan prediksi dari *lead time* yang lebih panjang akan meningkatkan ukuran *ensemble* serta dapat meningkatkan kemampuan (*skill*) prediksi dan keandalannya (*reliability*) (Kumar dan Hoerling, 2000, seperti dikutip dari Chen dkk, 2013). Namun di sisi lain, proses memasukkan prediksi dari *lead time* yang lebih panjang dapat juga menghasilkan penurunan kemampuan prediksi (Kumar dkk, 2011, seperti dikutip dari Chen dkk, 2013).

1.4 Chaos, Prediktabilitas Atmosfer, dan Prediksi Ensemble

PCN telah berkembang pesat terutama selama setengah abad belakangan ini dan mencapai tingkat kepercayaan yang tinggi di negara-negara industri maju. Ketika Wilhelm Bjerknes mencetuskan ide mengenai PCN kurang lebih seabad yang lalu, telah disadari bahwa terdapat dua syarat mutlak yang harus dipenuhi agar model cuaca numerik dapat memberikan prediksi yang akurat yaitu: (1) pengetahuan yang lengkap mengenai kondisi awal atmosfer, dan (2) pengetahuan yang lengkap mengenai seluruh proses yang terjadi di atmosfer yang dapat dituangkan ke dalam bentuk persamaan matematis.

Kita mengetahui bahwa secanggih apapun teknologi observasi atmosfer, data cuaca yang dihasilkan mengandung galat dan bagaimanapun rapatnya titik pengamatan tidak dapat menghasilkan resolusi yang cukup tinggi untuk menangkap semua gejala cuaca. Meskipun teknologi asimilasi data telah sedemikian rupa dibangun untuk mengatasi kekurangan-kekurangan ini, jelaslah syarat pertama tersebut di atas jelas tidak akan terpenuhi secara mutlak.

Ketika Jule Charney dan C.G. Rossby di tahun 1950-an berhasil mengimplementasikan PCN secara operasional, sempat muncul optimisme berlebihan mengenai sifat deterministik atmosfer yang dianggap telah terwakili oleh sistem persamaan gerak yang seluruhnya diketahui. Akan tetapi, tidak lama setelah itu Lorenz (1963) menunjukkan bahwa solusi persamaan non-linier dalam model cuaca numerik secara inheren akan memunculkan solusi bersifat *chaotic* karena perbedaan kecil di dalam nilai awal. Oleh karena itu, atmosfer tidak selalu *predictable* meskipun diatur oleh hukum-hukum fisika Newtonian yang deterministik. Keterbatasan model cuaca numerik ini secara cukup mengagumkan didemonstrasikan oleh Edward Lorenz, professor Meteorologi di Massachusetts Institute of Technology (MIT) pada waktu itu, menggunakan suatu sistem persamaan differensial biasa sebagai berikut

$$\left. \begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dz}{dt} &= xy - bz \end{aligned} \right\} \quad (1)$$

dengan σ, b, r adalah konstanta. Lorenz (1963) mendapatkan bahwa integrasi persamaan (18) terhadap waktu dengan $\sigma = 10$, $b = 8/3$ dan $r = 28$ menghasilkan solusi chaotic yang terkenal dengan pola atraktor Lorenz.

Meskipun mengandung ketidakpastian, PCN tetap memberikan informasi yang sangat berguna bagi manusia mengenai salah satu faktor penting yang menentukan keadaan masa depan. Hal yang paling penting adalah bagaimana mengkuantifikasi ketidakpastian tersebut sehingga resiko yang diakibatkan dapat diperkirakan dengan baik. Di berbagai pusat prediksi cuaca di dunia saat ini dikembangkan metoda prediksi ensemble yang dapat dilakukan dengan satu atau banyak (multi) model meskipun dasar ilmiahnya masih diperdebatkan. Prediksi ensemble menggunakan satu model didapatkan dengan cara memberikan nilai awal yang sedikit berbeda. Sekumpulan nilai awal dapat dibuat dengan memberikan suatu variasi terhadap nilai awal yang dihasilkan dari proses asimilasi data. Metoda ini akan menghasilkan sejumlah anggota (member) prediksi dengan keluaran yang berbeda. Statistik dari sejumlah keluaran model ini akan menentukan tingkat kemungkinan kebenaran prediksi.

1.4.1 Ensemble Forecast

Ensemble forecast atau prediksi ensemble adalah suatu aproksimasi dari prediksi probabilistik berdasarkan model meteorologi. *Ensemble Prediction System* (EPS) digunakan secara operasional dalam beberapa pusat prediksi cuaca dan iklim di dunia. Misalnya berdasarkan Molteni dkk (1996) pada *European Center for Medium-Range Weather Forecast* (ECMWF) atau Pellerin dkk (2003) pada *Meteorology Service of Canada* yang memproduksi keluaran ensemble secara rutin. Prediksi-prediksi ini telah dibuktikan kegunaannya dalam proses pengambilan keputusan (Callado, 2013). Terdapat beberapa

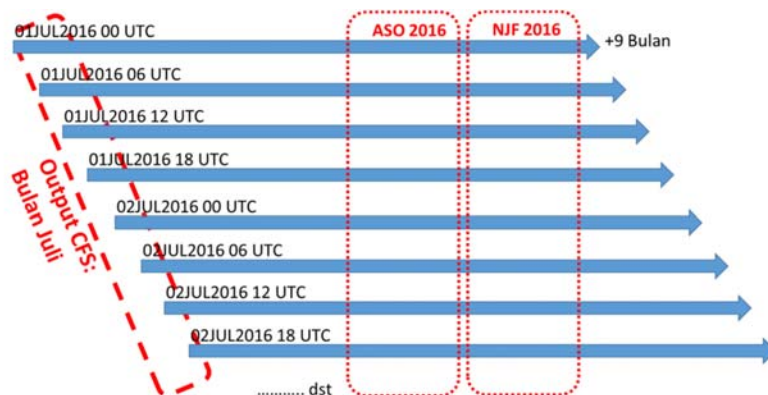
metode dalam prediksi ensemble yaitu *multi-model SuperEnsemble* Khrisnamurti dkk (1999) dan *time-lagged ensemble* (Hoffman dan Kalnay, 1983). *Multi-model SuperEnsemble* merupakan salah satu metode kuat untuk membangun EPS. Dengan menggunakan keluaran banyak model yang ideal untuk mendapat suatu estimasi gabungan dari parameter cuaca. Adapun *time-lagged ensemble* merupakan metode yang menggunakan perbedaan *initial time-lag* dari prediksi sebagai bagian dari *member*-nya. Adapun *time-lagged ensemble* memiliki kekurangan yaitu semakin jauh *lag* waktu dari prediksi, maka *skill* hasil prediksi menjadi kurang baik (Callado dkk, 2013).

1.4.2 Ensemble Mean

Pada hakikatnya, penggunaan metode *time-lag ensemble* memiliki salah satu kekurangan yaitu *lack of scientific motivation* karena metode ini hanya didasarkan pada perbedaan initial kondisi (Hoffman dan Kalnay, 1983). Sehingga, tidak ada keterkaitan secara fisis antara model keluaran periode *developing* terhadap periode *testing*. Dengan demikian, diperlukan cara untuk mengetahui hubungan antar member pada periode *developing* dan *testing* secara statistik. Adapun untuk mencari hubungan antara kedua periode tersebut, digunakan metode *ensemble mean* yang dapat digunakan untuk melihat persebaran *members* terhadap rata-ratanya (McElroy, 2006). *Ensemble mean* adalah metode perata-rataan sederhana dari nilai suatu parameter yang dikeluarkan oleh seluruh *members* keluaran CFSv2.

1.4.3 Time Lag Ensemble

Konsep *Time lag ensemble* berbeda dengan *Multi-Model Ensembl* (MME), dimana *member* I pada periode *developing* tidak memiliki keterkaitan fisis dengan periode *testing*. *Time Lag-ensemble* merupakan keadaan yang sangat acak pada *initial condition*-nya sehingga memungkinkan prediksi yang lebih variatif pada suatu kejadian. Hal ini tentunya berbeda dengan konsep MME yang bertujuan untuk menghitung error baik pada inisial kondisi ataupun *model forecast*-nya (Johnson and Swinbank). Konsep *time lag ensemble*



Gambar 1 Ilustrasi Skema Lag-Time Ensembl

1.4.4 Forecast Verification

Forecast Verification atau verifikasi prediksi merupakan proses menilai kualitas prediksi yang dilakukan. Proses penilaian kualitas prediksi ini terkadang disebut sebagai validasi atau evaluasi dari prediksi. Penilaian dari kualitas prediksi ini melibatkan distribusi data prediksi dengan data observasi dan hubungan antar keduanya. Idealnya hubungan antar data observasi dengan data prediksi menunjukkan nilai yang cukup kuat biasanya terlihat dari distribusi data observasi maupun data prediksi.

Verifikasi prediksi dilakukan biasanya untuk memantau prediksi yang dilakukan secara operasional. Pemantauan ini dilakukan dengan melakukan pemantauan nilai trend dari model prediksi dalam rentang waktu operasional riset. Analisis verifikasi secara statistik juga dapat membantu dalam pengukuran kualitas dari hasil prediksi. Analisis verifikasi secara statistik juga dapat melihat umpan balik dari data prediksi terhadap data observasi, kemudian dapat menilai dan mengevaluasi kekurangan kinerja prediksi yang dilakukan secara statistik. Hal ini dilakukan dengan harapan kinerja prediksi di kemudian hari menghasilkan prediksi dengan kualitas yang lebih baik.

1.4.4.1 The Brier Score

Mengingat dimensi probabilitas verifikasi yang umumnya tinggi melibatkan prediksi probabilistik dari suatu dataset, sehingga biasanya digunakan verifikasi prediksi menggunakan pengukuran secara scalar. Pengukuran scalar ini tentunya menghasilkan gambaran kinerja prediksi yang kurang lengkap. Pengukuran nilai akurasi secara skalar untuk verifikasi prediksi probabilistik kejadian yang terjadi. Sejauh ini pengukuran akurasi prediksi probabilistik ini menggunakan Brier Score. Konsep Brier Score pada dasarnya adalah nilai *mean squared error* dari prediksi probabilistik dengan mempertimbangkan observasi $o_1 = 1$ jika terjadi even, dan observasi $o_2 = 0$ jika tidak terjadi even. Skor merata-ratakan nilai kuadrat berbeda antara pasangan prediksi probabilistik dengan nilai observasi. Penentu Brier Skor menggunakan persamaan berikut ini:

$$BS = \frac{1}{n} \sum_{k=1}^n (y_k - o_k)^2 \quad (2)$$

Dengan indeks k sebagai n jumlah pasangan prediksi-observasi. Membandingkan Brier Score dengan persamaan *mean-squared-error*, seperti halnya pengukuran akurasi dengan persamaan *mean-squared-error*, Brier Score berorientasi negatif, dengan prediksi sempurna yang menunjukkan $BS = 0$. Prediksi yang kurang akurat mendapat nilai Brier yang lebih tinggi, namun karena perkiraan individu dan pengamatan keduanya dibatasi oleh nol dan satu, skor dapat mengambil nilai hanya pada kisaran $0 \leq BS \leq 1$.

BAB 2

BAB 2 SD (Statistical Downscaling)

2.1 Pengenalan SD (*Statistical Downscaling*)

Metode *Statistical Downscaling* (SD) pertama kali diterapkan dalam ramalan cuaca *short-range* dan kemudian diadaptasi untuk *large-horizon*, termasuk aplikasi prediksi musiman dan perubahan iklim. Penelitian-penelitian yang menggunakan metode SD sebagian besar membahas mengenai aplikasi perubahan iklim dan prediksi cuaca, sehingga persyaratan dan hipotesis dari skala temporal yang berbeda harus dianalisis dengan cermat dalam setiap kasus. Misalnya masalah asumsi stasioner metode SD sangat penting untuk aplikasi perubahan iklim, namun kurang penting dari perspektif peramalan musiman.

2.2 Kerangka Klasifikasi Untuk Metode SD

Penerapan teknik SD dalam penelitian iklim dan cuaca sangat luas, berdasarkan teknik statistik tertentu yang digunakan untuk membangun hubungan antara variabel skala besar dan variabel skala lokal (fungsi transfer, analog dan *weather typing-weather generator*). Penulis lain membuat klasifikasi yang berbeda berdasarkan pendekatan downscaling (berdasarkan sifat prediktor yang digunakan dalam fase *training*, prediksi atau analisis), yang mengarah ke dua kategori penting: *Model Output Statistics* (MOS) dan *Perfect Prognosis* (PP).

2.2.1 Teknik SD (*Statistical Downscaling*)

Metode SD dapat diklasifikasikan ke dalam kategori yang berbeda, diantaranya:

- *Transfer Function*, biasanya didasarkan pada regresi linier atau nonlinier, untuk menyimpulkan hubungan antara prediktor dan prediktan dalam skala besar. Metode ini bersifat "generatif" dalam arti bahwa metode ini menghasilkan model statistik (atau formula), yang parameternya sesuai dengan data historis yang telah ada. Berbagai versi *multiple* regresi linier tidak hanya untuk digunakan untuk variabel suhu tetapi juga untuk variabel non-gaussian, seperti presipitasi.

- *Analog dan weather typing*, biasanya didasarkan pada kejadian lokal yang sesuai dengan satu rangkaian *historical analogs* (*nearest neighbors*) dari prediktor (biasanya ditunjukkan oleh variabel sirkulasi berskala besar dalam domain skala sinoptik). Perangkat analog kadang-kadang diganti dengan pre-klasifikasi prediktor menjadi variabel cuaca yang terbatas, diperoleh sesuai dengan kesamaan sinoptik. Metode ini biasanya tidak digunakan secara umum, karena terdiri dari prosedur algoritmik yang dikalibrasi / disesuaikan dengan data historis yang tersedia selama fase training. Contoh populer dari teknik ini adalah metode *analog* (Lorenz, 1969; Zorita dan von Storch, 1999; Goodess dan Palutikof, 1998; Imbert dan Benestad, 2005; Cheng et al., 2011), versi stokastik yang dikenal sebagai *analog resampling* (Beersma dan Buishand, 2003), dan implementasi dari teknik *weather typing* (lihat, misalnya Gutiérrez et al., 2004; Gutiérrez et al., 2005; Pagé et al., 2009).
- *Weather generator*, model stokastik yang dapat membuat *time series* data secara acak dengan sifat statistik yang serupa dari pada deret waktu yang diamati (rata-rata, variabilitas, frekuensi data ekstrem, panjang musim kering dan basah, dll.). Berbeda dengan kategori sebelumnya, metode ini hanya dapat mengenalkan beberapa karakteristik yang diinginkan dari struktur temporal atau spasial yang diamati (misalnya autokorelasi temporal) ke dalam model simulasi. Untuk tujuan tertentu dari SD, beberapa parameter *weather generator* (misalnya probabilitas transisi HMM) dimodifikasi sesuai dengan keluaran GCM (lihat Robertson et al., 2004), atau sesuai dengan keluaran dari metode SD yang berjalan pada skala bulanan atau musiman (misalnya faktor delta, CCA, dll.) (Lihat Robertson dkk, 2004; Wilks dan Wilby, 1999; Feddersen dan Andersen, 2005; Kilsby et al., 2007; Hirschi et al., 2012, untuk implementasi teknik yang berbeda).

Terlepas dari kategori ini, beberapa algoritma koreksi bias (faktor skala, pemetaan nilai quantile, dll) telah diperkenalkan untuk mengkalibrasi secara statistik model keluaran sesuai dengan sifat statistik dari rangkaian yang diamati. Teknik ini bisa dipertimbangkan saat ini sebagai kelompok teknik SD tambahan. Selain itu, ada juga teknik hibrida yang dapat menggabungkan metode dari kelas yang berbeda menjadi teknik *single multi-step* (misalnya langkah pertama pemilihan analog dan langkah kedua regresi pada rangkaian analog). Dari perspektif statistik, klasifikasi sederhana teknik downscaling di atas dapat juga ditetapkan dalam bentuk metode generatif atau non-generatif, bergantung pada apakah mereka bergantung pada model matematis.

2.2.2 Pendekatan Metode SD

Penerapan teknik downscaling statistik untuk output model global biasanya memerlukan dua fase. Pada fase training, parameter model (atau algoritma) dipasang pada data (atau dikalibrasi) dan dilakukan *cross-validated* menggunakan data periode historis yang representatif (biasanya beberapa dekade) dengan data prediktor dan prediktan yang ada. Pada fase downscaling, yang umum untuk semua metode SD, prediktor yang diberikan oleh keluaran output model global dimasukkan ke dalam model (atau algoritma) untuk mendapatkan nilai SD lokal yang sesuai untuk prediksi. Menurut pendekatan yang diikuti dalam fase training, teknik SD biasanya dikelompokkan menjadi dua kategori yang berbeda seperti berikut:

- *Prefect Prognosis (PP)*: Pada fase training dengan pendekatan PP, model statistik dikalibrasi dengan menggunakan data observasi untuk prediktor dan prediktan. Data observasi diambil dari dataset data reanalisis yang berbeda. Karena model global yang digunakan berbeda dalam fase training dan fase downscaling, variabel sirkulasi skala besar merepresentasikan model global dengan baik biasanya dipilih sebagai prediktor dalam pendekatan ini. Variabel secara langsung dipengaruhi oleh parameterisasi dan model orografi (misalnya presipitasi, indeks ketidakstabilan, dll.) tidak dianggap sebagai prediktor yang sesuai dalam pendekatan ini. Oleh karena itu, salah satu tugas yang paling banyak memakan waktu untuk metode ini adalah pemilihan kombinasi prediktor yang sesuai untuk setiap prediktan pada studi kasus tertentu. Tahapan ini biasanya dilakukan dengan melakukan *cross-validating* dari berbagai konfigurasi dengan menggunakan nilai validasi standar: ukuran akurasi, nilai kemiripan distribusi, variabilitas data, nilai tren, dll. sejumlah aspek perlu diperhatikan secara hati-hati. Hal yang harus diperhatikan dalam pemilihan prediktor dalam pendekatan ini diantaranya: 1) Prediktor harus memperhitungkan sebagian besar variabilitas dalam prediksi, 2) hubungan antara prediktor dan prediktan harus stabil baik secara temporer ataupun stasioner dan 3) prediktor skala besar harus direproduksi secara realistis oleh model iklim global.
- *Model Output Statistics (MOS)*: Dalam pendekatan MOS, prediktor diambil dari model global (atau regional) yang sama untuk fase training dan downscaling. Dengan demikian, metode ini biasanya bekerja dengan prediksi (interpolasi lokal) dan variabel pengamatan (prediktor tunggal). Misalnya, dalam pendekatan MOS, presipitasi lokal biasanya diturunkan dari perkiraan presipitasi model langsung.

Salah satu kekurangan utama pendekatan PP adalah data reanalisis tidak selalu memberikan variabel prediktor yang representatif secara sempurna, oleh karena itu ketidakpastian reanalisis harus dianalisis dengan hati-hati sebagai sumber ketidakpastian tambahan, terutama di daerah tropis. Selain itu, pendekatan ini sensitif terhadap bias model global sistematis yang khas, yang tidak terwakili dalam data reanalisis. Jadi, pra-pengolahan yang tepat (menghilangkan bias model atau standarisasi bidang reanalisis dan model) diperlukan untuk pendekatan ini agar mendapatkan hasil yang berarti.

Tabel 1 Diagram konseptual kerangka klasifikasi untuk metode downscaling statistik sesuai dengan teknik statistik yang berbeda (Tek.) Dan pendekatan (Appro.), Yang juga menunjukkan sifat

Diktat Pelatihan SD
(Statistical Downscaling)

deterministik atau stochastic dari teknik dan deret waktu (event wise) atau karakter distribusi kalibrasi.

Appro. \ Tech.		Generative		non-Generative	
		Deterministic	Stochastic	Deterministic	Stochastic
PP	Eventwise	Regression, Neural Nets.	GLMs	Analogs, weather types	Analog resampling
	Distribution	Regression on PDF parameters			
MOS	Eventwise	Regression, Neural Nets.	GLMs	Analogs	Analog resampling
	Distribution	Bias correction, parametric q-q map	Nonhomogeneous HMM	q-q map	

BAB 3 CONSTRUCTED ANALOGUE DAN STATISTICAL BIAS CORRECTION

3.1 *Statistical Bias Correction*

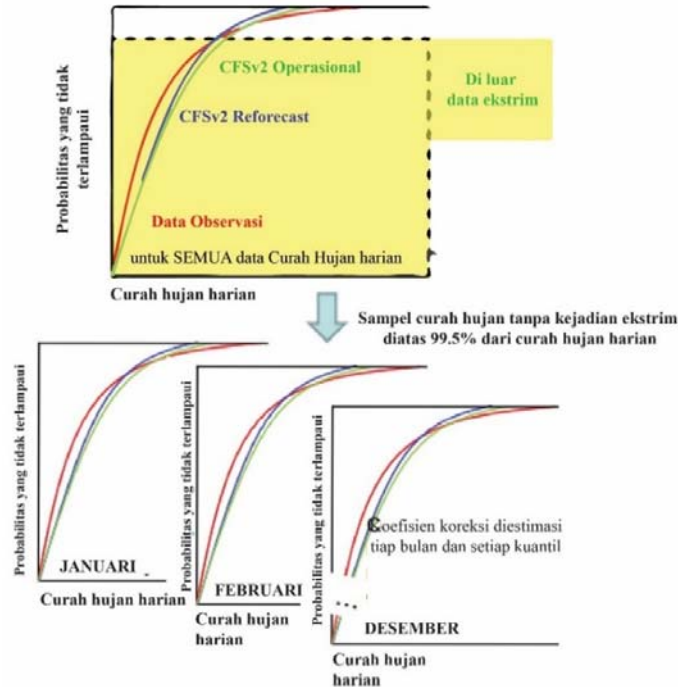
Berdasarkan (Fukami dan Inomata, 2011) terdapat beberapa hal yang perlu diperhatikan dalam penggunaan metode *statistical bias correction* yakni sebagai berikut. Keluaran CFSv2 dibagi tiap bulan. Dikarenakan penelitian ini fokus pada prediksi *onset monsun* dan AMH, maka faktor koreksi yang akan dihitung adalah bulan Oktober, November, dan Desember saja.

Langkah awal penggunaan metode *Statistical Bias Correction* sebagai metode *statistical downscaling* CFSv2 kali ini adalah dengan mencari rasio untuk setiap kuantil (α_q) diantara data observasi curah hujan (pada penelitian kali ini akan digunakan data TRMM sebagai *secondary data*) ($P_{trmm_{m,q}}$) dengan estimasi curah hujan berdasarkan CFSv2 *Reforecast* ($CFSv2_{re_q}$). α_q dianggap sebagai faktor koreksi untuk setiap kuantil dan sebagai faktor pengali untuk nilai-nilai estimasi CFSv2 pada periode *testing* CFSv2 dan operasional di kuantil yang sama ($CFSv2_{pred_q}$) dan nilai curah hujan yang telah dikoreksi (P_{pred_q}) yang diperoleh. Adapun faktor koreksi tiap *quantile* adalah sebagai berikut.

$$\alpha_q = \frac{P_{trmm_{m,q}}}{CFSv2_{re_{m,q}}} \quad (3)$$

Setelah mendapatkan data faktor koreksi dan bantar nilai curah hujan per-kuantil, faktor koreksi akan digunakan untuk mengoreksi curah hujan pada periode testing dengan persamaan sebagai berikut.

$$P_{pred_{m,q}} = \alpha_{m,q} \times CFSv2_{pred_{m,q}} \quad (4)$$



Gambar 2 Contoh penggunaan statistical bias correction pada *downscaling* data CFSv2 di luar fenomena ekstrim terhadap curah hujan TRMM diadaptasi dari (Fukami dan Inomata, 2011)

3.2 Constructed Analogue

Metoda analog telah umum digunakan dalam prediksi cuaca dan musim sejak masa Lorenz (1969), walaupun dengan hasil yang kurang memuaskan. Hal ini terutama disebabkan karena analog dipilih dari database pola atmosfer yang kasar (van den Dool, 1994). Sementara penggunaan metode analog untuk proses *downscaling* sendiri dimulai oleh Zorita dkk. (1995). Salah satu tujuan dari pengenalan metode ini dalam proses *downscaling* adalah untuk memperoleh statistik iklim lokal yang konsisten dengan kondisi atmosfer skala luas (Zorita dan von Storch, 1998). Ketika metode ini diterapkan untuk men-*downscale* iklim lokal di masa depan, database yang digunakan idealnya harus dapat mencakup seluruh pola atau kondisi atmosfer yang mungkin muncul di masa depan. Oleh karenanya, keberhasilan aplikasi metode ini di suatu wilayah akan sangat tergantung dari ketersediaan data (pengamatan) yang cukup panjang.

Metode analog merupakan metode *downscaling* yang relatif mudah diaplikasikan, dengan kelebihan utama: metode ini menggunakan pola iklim dari data pengamatan sehingga struktur spasial dari iklim lokal terwakilkan dengan baik di hasil simulasi. Oleh karena metode ini meng"analogikan" pola iklim skala lokal dengan suatu kondisi/pola iklim skala luas, hal penting yang perlu ditentukan sebelumnya adalah variabel iklim skala luas mana yang terbaik untuk mewakili variabel iklim lokal yang akan di analogikan. Timbal dan McAvaney (2001) mengajukan kriteria prediktor untuk metode analog:

- (1) secara realistis mampu disimulasikan oleh model sirkulasi global,

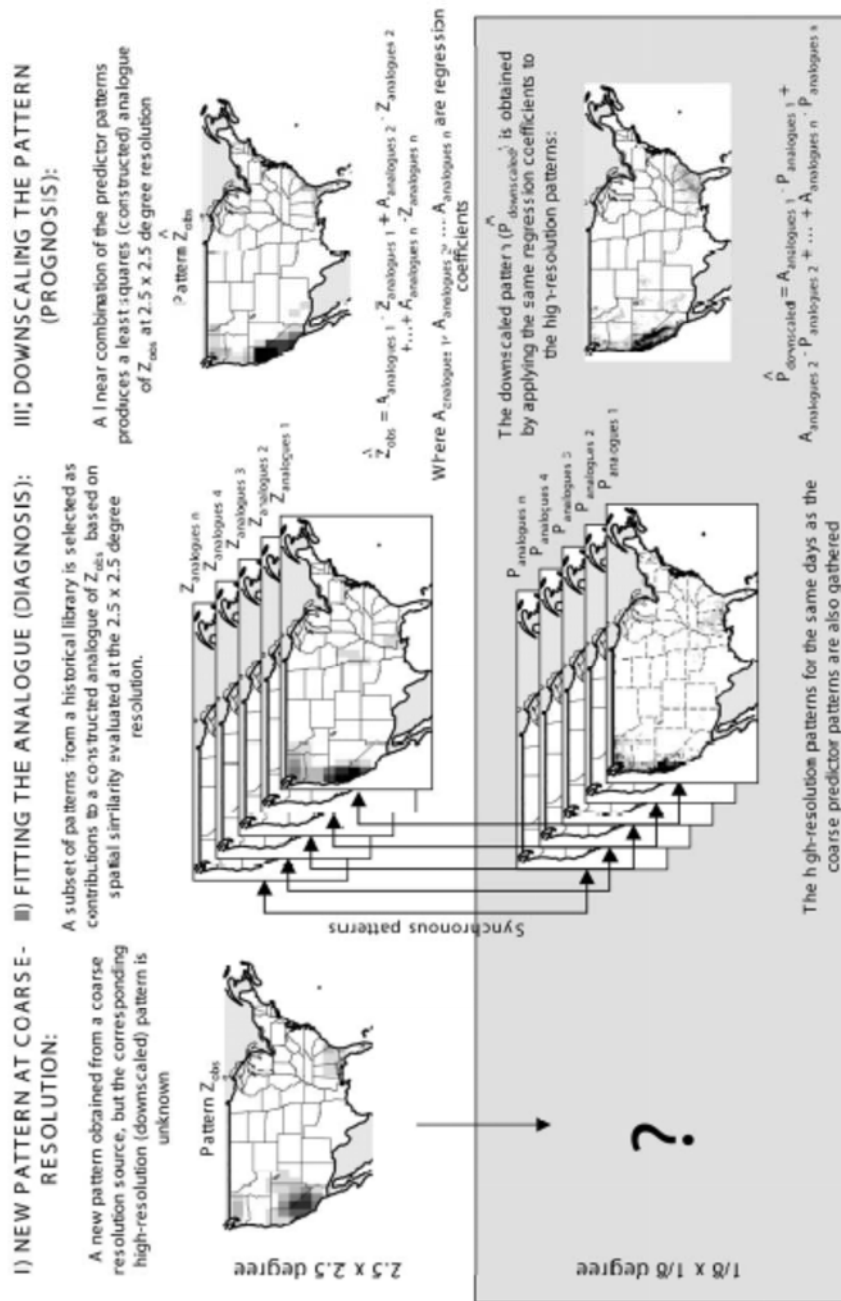
- (2) memiliki skill prediktif yang kuat terhadap variabel prediktan, dan
- (3) memiliki keterkaitan fisis.

Berdasarkan kriteria tersebut, Hidalgo dkk. (2008) membagi dua pendekatan yang digunakan dalam penentuan prediktor: metode analog secara tidak-langsung (*indirect analogue method*), yang menggunakan variabel sirkulasi atmosfer skala luas sebagai prediktor, dan metode analog secara langsung (*direct analogue method*) yang menggunakan variabel skala luas yang sama dengan variabel lokal (prediktan).

Hidalgo dkk. (2008) menunjukkan bahwa *indirect method* pada umumnya disertai dengan penggunaan analisis EOF (*empirical orthogonal function*) (seperti dalam Zorita dan von Storch, 1998). Tujuannya untuk memfilter *noise* dari data dan juga mengurangi derajat kebebasan (d.o.f., *degree of freedom*) dari variabel prediktor yang dipilih. Konsep mengenai d.o.f dari variabel sirkulasi atmosfer sangat penting dalam metode analog ini. Lorenz (1969) telah menunjukkan bahwa kemungkinan ditemukannya suatu analog dari pola iklim skala hemisfer dengan kualitas yang dapat diterima sangat kecil, oleh karena tingginya d.o.f sirkulasi atmosfer dan pendeknya rentang waktu pengamatan yang tersedia. Van den Dool (1994) bahkan memberikan estimasi bahwa butuh hingga 10^{30} tahun untuk menemukan dua pola atmosfer yang identik.

Untuk mengatasi masalah ini, Hidalgo dkk. (2008) mengajukan metode *direct analogue*, yaitu menggunakan variabel yang sama, baik sebagai prediktor (skala luas) maupun prediktan (skala lokal). Kontras dengan metode yang diajukan oleh antara lain Zorita dan von Storch (1998), yang menggunakan metode *indirect analogue* dan analisa EOF untuk mengurangi derajat kebebasan, Hidalgo dkk. (2008), mengikuti van den Dool (1994), mencari sekumpulan analog-analog yang berpotensi, lalu membangun satu analog berdasarkan hubungan linier dari sekumpulan analog-analog tersebut (Gambar II.1). Metode ini disebut sebagai *constructed analogue*, untuk membedakannya dengan metode sebelumnya yang disebut *single best analogue*.

Faktor lainnya yang mempengaruhi kemampuan *downscaling* metode analog adalah penetapan ukuran kemiripan (*similarity*). Oleh karena proses pencarian analog suatu pola iklim di suatu waktu di masa mendatang didasarkan pada pencarian pola atmosfer yang mirip di database, kemampuan *downscaling* dengan metode ini akan sangat bergantung bagaimana kemiripan (*similarity*) antar pola tersebut didefinisikan dalam proses pencarian analog (Zorita dan von Storch, 1998).



Gambar 3 Skema metode *constructed analog* yang diujukan oleh Hidalgo dkk. (2008)

3.2.1 Metode Pengolahan Pendekatan CA (*Constructed Analogue*)

Dalam penelitian ini diajukan dua pendekatan dalam metode CA yang digunakan: *single window* dan *multiple-window*. Pendekatan *single windows* dilakukan dalam tahap penentuan prediktor terbaik dari kandidat prediktor yang dipilih. Selanjutnya pendekatan *multiple-window* diterapkan untuk memperoleh suatu skema CA yang *robust* untuk seluruh wilayah dalam kajian ini.

Metode CA yang diterapkan dalam penelitian ini sendiri secara garis besar mengikuti metode yang diajukan oleh Zorita dan von Storch (1998), van den Dool (1994), dan Hidalgo dkk. (2008). Metode CA pada umumnya dibagi menjadi dua proses:

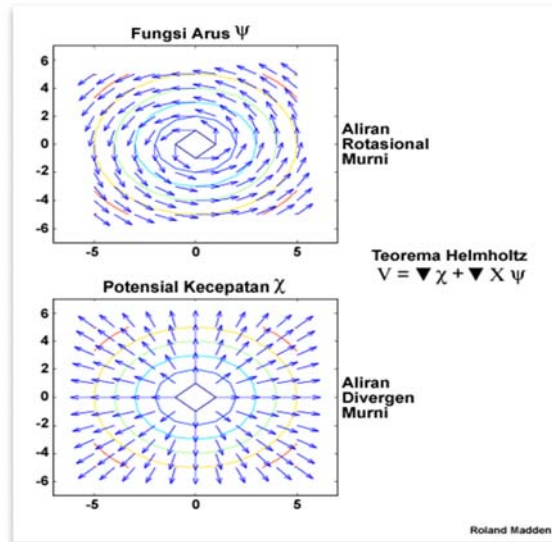
1. Proses diagnosis: mencari sekumpulan analog dari prediktor di database yang tersedia.
2. Proses prognosis: membentuk analog dari prediktor di waktu target, berdasarkan sekumpulan analog prediktor yang ditemukan sebelumnya.

Dalam kajian ini dilakukan beberapa modifikasi dari metode CA pada umumnya. Modifikasi ini dilakukan dalam proses diagnosis (sub-bab III.2.2) dan penerapan skema *multi window* dalam metode CA (sub-bab III.2.4). Tahapan-tahapan dalam kajian ini akan dijelaskan lebih lanjut dalam sub-bab-sub-bab berikut.

3.2.2 Pendekatan Single Window: Penentuan Prediktor

Pemilihan prediktor merupakan tahapan yang sangat menentukan kesuksesan metode analog. Dalam penelitian ini digunakan pendekatan "*indirect analogue downscaling method*" yang berarti menggunakan variabel prediktor dan prediktan yang berbeda. Dalam kajian ini digunakan sirkulasi atmosfer lapisan atas untuk men-*downscale* variabel di permukaan (curah hujan). Variabel sirkulasi atmosfer yang dipilih sebagai prediktor dalam kajian ini adalah variabel yang merepresentasikan medan angin horizontal. Alasan pemilihan medan angin sebagai prediktor adalah karena medan angin merupakan variabel yang menentukan fase-fase evolusi monsun.

Tiga variabel medan angin yang dicoba adalah: medan angin zonal (U850), medan fungsi arus (PSI), dan medan potensial kecepatan (PHI), ketiganya di lapisan 850 mb. Pemilihan U850 sebagai prediktor karena variabel ini banyak digunakan untuk merepresentasikan indeks-indeks monsun. Variabel U850 diambil langsung dari keluaran model CFS-R. Sementara PSI dan PHI adalah kuantitas sirkulasi atmosfer yang diturunkan dari variabel angin zonal dan angin meridional keluaran model CFS-R. PSI dan PHI masing-masing merepresentasikan komponen rotasional dan divergen dari medan angin horizontal (Gambar 7).



Gambar 4 Ilustrasi dari PSI dan PHI (COMET Program, 2002)

Berdasarkan teorema Helmholtz hubungan antara medan angin dengan PHI dan PSI dapat dinyatakan dengan persamaan:

$$\vec{V} = \vec{k} \times \nabla\psi - \nabla\chi \quad (5)$$

dimana ψ adalah fungsi arus (PSI) dan χ adalah potensial kecepatan (PHI). Hubungan antara komponen-komponen angin horizontal dengan kuantitas PHI dan PSI berikutnya dapat dinyatakan dengan persamaan:

$$u = -\frac{\partial\psi}{\partial y} - \frac{\partial\chi}{\partial x} \quad (6)$$

dan

$$v = \frac{\partial\psi}{\partial x} - \frac{\partial\chi}{\partial y} \quad (7)$$

dimana u dan v adalah komponen angin zonal dan meridional. Ketiga variabel tersebut (U850, PHI, dan PSI) digunakan dalam eksperimen CA sebagai prediktor. Selanjutnya prediktor yang terbaik dipilih dengan membandingkan hasil ketiga eksperimen tersebut dengan data pengamatan di lapangan.

3.2.3 Proses Diagnosis

Bagian ini melibatkan proses pencocokan pola prediktor di waktu target terhadap pola prediktor di database. Tujuan dari proses diagnosis adalah menemukan sekelompok analog terbaik (*subset of*

suitable analogues). Dalam penelitian ini diajukan beberapa modifikasi dari tahapan diagnosis oleh Hidalgo dkk. (2008):

1. Penerapan analisis EOF untuk mengurangi d.o.f pada medan sirkulasi atmosfer yang digunakan sebagai prediktor (Zorita dan von Storch, 1998). Database prediktor $F(u)$ dan prediktor di waktu target $F(t)$ direduksi menjadi komponen spasial (c_k) dan komponen temporal (a_k):

$$F(u) = \sum_{k=1}^M c(u)_k a(u)_k \quad (8)$$

dan

$$F(t) = \sum_{k=1}^M c(t)_k a(t)_k \quad (9)$$

dengan M adalah jumlah komponen yang signifikan (*principal components*).

2. Proses pencarian analog dilakukan dengan menggunakan metode *Cosine Similarity* (Garcia, 2006). *Cosine Similarity* digunakan untuk menghitung derajat kemiripan antara vektor $\vec{a}(u)$ dan $\vec{a}(t)$ dengan persamaan:

$$S(u) = \frac{\vec{a}(u) \cdot \vec{a}(t)}{|\vec{a}(u)| |\vec{a}(t)|} \quad (10)$$

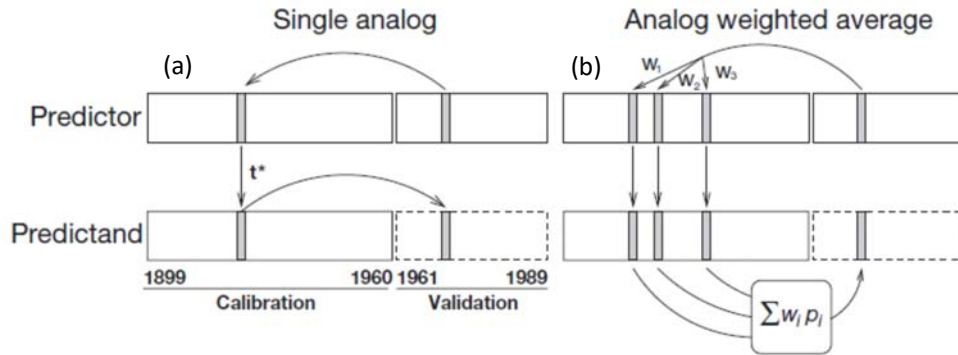
Untuk setiap waktu target t , sekumpulan analog dari prediktor di database dan prediktan yang berpasangan dengannya dipilih berdasarkan nilai similaritas $S(u)$.

3.2.4 Proses Prognosis

Proses prognosis bertujuan untuk membentuk satu analog curah hujan (*constructed analogue*) di setiap waktu target t , berdasarkan sekelompok (*subset*) analog terbaik yang diperoleh dari proses diagnosis. Dalam kajian ini digunakan metode perata-rataan berbobot dalam proses prognosis (Fernández dan Sáenz, 2003). Pertama, bobot (W) masing-masing anggota *subset* analog prediktor ditentukan berdasarkan koefisien korelasi dan RMSE-nya terhadap prediktor target. Selanjutnya bobot ini digunakan dalam pembentukan *constructed analogue* prediktan di waktu target t :

$$Z_{CA}(t) = \sum_{i=1} W_i P_i \quad (11)$$

dengan P adalah *subset* analog prediktan yang berpasangan dengan *subset* analog prediktor dan Z adalah *constructed analogue* dari prediktan. Ilustrasi sederhana dari proses diagnosis dan prognosis ditunjukkan dalam Gambar 8.



Gambar 5 Ilustrasi proses diagnosis (pencarian analog) dan prognosis untuk (a) *single best analogue* dan (b) *constructed analogue*. Dalam ilustrasi ini digunakan metode perata-rataan berbobot dalam proses prognosisnya (Fernández dan Sáenz, 2003)

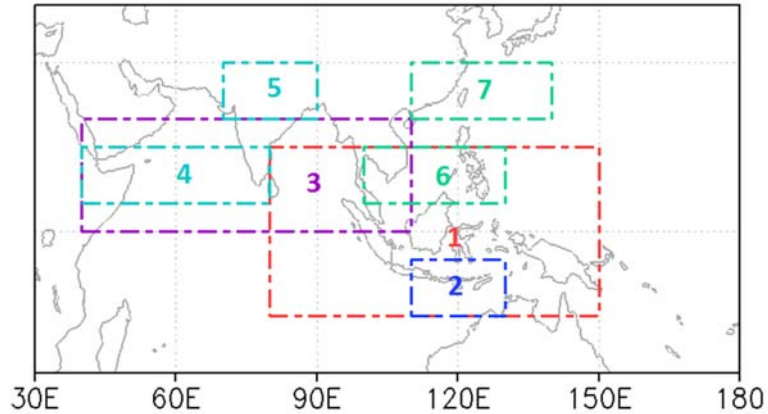
3.2.5 Pendekatan Multiple-window

Pendekatan *multiple-window* diajukan dengan harapan mendapatkan suatu skema yang *robust* untuk seluruh wilayah kajian ini. Dalam tahapan ini di pilih tujuh buah *windows* yang sirkulasi atmosfernya diasumsikan mempengaruhi curah hujan di wilayah Indonesia.

Window-window yang dipilih adalah wilayah-wilayah yang didefinisikan sebagai wilayah monsun oleh kajian-kajian indeks monsun ditambah wilayah benua maritim Indonesia. Wilayah-wilayah yang dipilih sebagai *domain window* adalah:

- Benua maritim Indonesia: 80° – 150° BT dan 15° LS – 15° LU
- Wilayah indeks monsun Australia: 110° – 130° BT dan 15° – 5° LS (Kajikawa dkk., 2010)
- Wilayah indeks monsun definisi Webster dan Yang (WYMI): 40° – 110° BT dan EQ – 20° LU (Webster dan Yang, 1992)
- Wilayah monsun India: 40° – 80° BT dan 5° – 15° LU serta 70° – 90° BT dan 20° – 30° LU (Wang dkk., 2001)
- Wilayah monsun Pasifik Barat Daya : 100° – 130° BT dan 5° – 15° LU serta 110° – 140° BT dan 20° – 30° LU (Wang dkk., 2001)

Ke-tujuh window tersebut ditunjukkan oleh Gambar 9.



Gambar 6 Domain windows yang digunakan dalam tahap diagnosis: (1) benua maritim, (2) wilayah monsun Australia, (3) wilayah indeks monsun definisi Webster dan Yang (WYMI), (4 dan 5) wilayah monsun India, serta (6 dan 7) wilayah monsun Pasifik Barat Daya. Untuk eksperimen penentuan prediktor, domain yang digunakan hanya benua maritim.

Metode CA diterapkan pada tujuh windows tersebut, sehingga masing-masing window akan menghasilkan satu CA curah hujan tersendiri: $\{CH_1(t), CH_2(t), \dots, CH_7(t)\}$. Selanjutnya ensemble dari ketujuh CA curah hujan tersebut diperoleh dengan menggunakan tiga metode berbeda:

1. Metode *multiple linear regression*. Dengan menggunakan metode *multiple linear regression*, dimana CA curah hujan tiap-tiap window sebagai variabel bebas dan curah hujan pengamatan (CH_{obs}) sebagai variabel terikat, akan diperoleh koefisien regresi $\{B_1, B_2, \dots, B_7\}$ dan konstanta regresi C . Ensemble dari CA curah hujan (CH_{ENS}) selanjutnya dapat dihitung dengan:

$$CH_{ENS}(t) = \sum_{n=1}^7 B_n CH_n(t) + C \quad (12)$$

2. Metode *stepwise regression*. Metode ini mirip dengan metode sebelumnya. Perbedaannya dengan metode ini tidak semua variabel-variabel bebas dimasukkan dalam perhitungan ensemble, hanya variabel-variabel bebas yang memenuhi kriteria statistik tertentu yang dimasukkan.
3. Metode perata-rataan berbobot. Pertama, bobot CA curah hujan masing-masing window ditentukan berdasarkan koefisien korelasi dan RMSE-nya terhadap CH_{obs} sehingga diperoleh $\{W_1, W_2, \dots, W_7\}$. CH_{ENS} selanjutnya dapat dihitung dengan

$$CH_{ENS}(t) = \sum_{n=1}^7 W_n CH_n \quad (13)$$

BAGIAN II

BAB 4

BAB 4 PENGENALAN MATLAB

Matlab merupakan bahasa canggih untuk komputansi teknik. Matlab merupakan integrasi dari komputansi, visualisasi dan pemograman dalam suatu lingkungan yang mudah digunakan, karena permasalahan dan pemecahannya dinyatakan dalam notasi matematika biasa. Kegunaan Matlab secara umum adalah untuk :

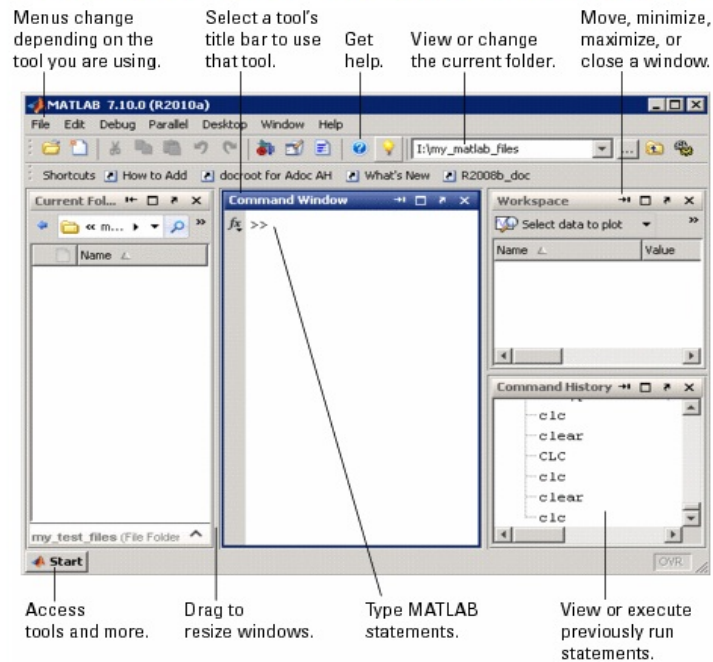
- Matematika dan Komputansi
- Pengembangan dan Algoritma
- Pemodelan, simulasi dan pembuatan prototype
- Analisa Data, eksplorasi dan visualisasi
- Pembuatan aplikasi termasuk pembuatan *graphical user interface*

Matlab adalah sistem interaktif dengan elemen dasar array yang merupakan basis datanya. Array tersebut tidak perlu dinyatakan khusus seperti di bahasa pemograman yang ada sekarang. Hal ini memungkinkan anda untuk memecahkan banyak masalah perhitungan teknik, khususnya yang melibatkan matriks dan vektor dengan waktu yang lebih singkat dari waktu yang dibutuhkan untuk menulis program dalam bahasa C atau Fortran. Untuk memahami matlab, terlebih dahulu anda harus sudah paham mengenai matematika terutama operasi vektor dan matriks, karena operasi matriks merupakan inti utama dari matlab. Pada intinya matlab merupakan sekumpulan fungsi-fungsi yang dapat dipanggil dan dieksekusi. Fungsi-fungsi tersebut dibagi-bagi berdasarkan kegunaannya yang dikelompokan didalam toolbox yang ada pada matlab. Untuk mengetahui lebih jauh mengenai toolbox yang ada di matlab dan fungsinya anda dapat mencarinya di website <http://www.mathworks.com>, atau anda dapat membuka cd dokumentasi matlab.

4.1 Lingkungan Kerja Matlab

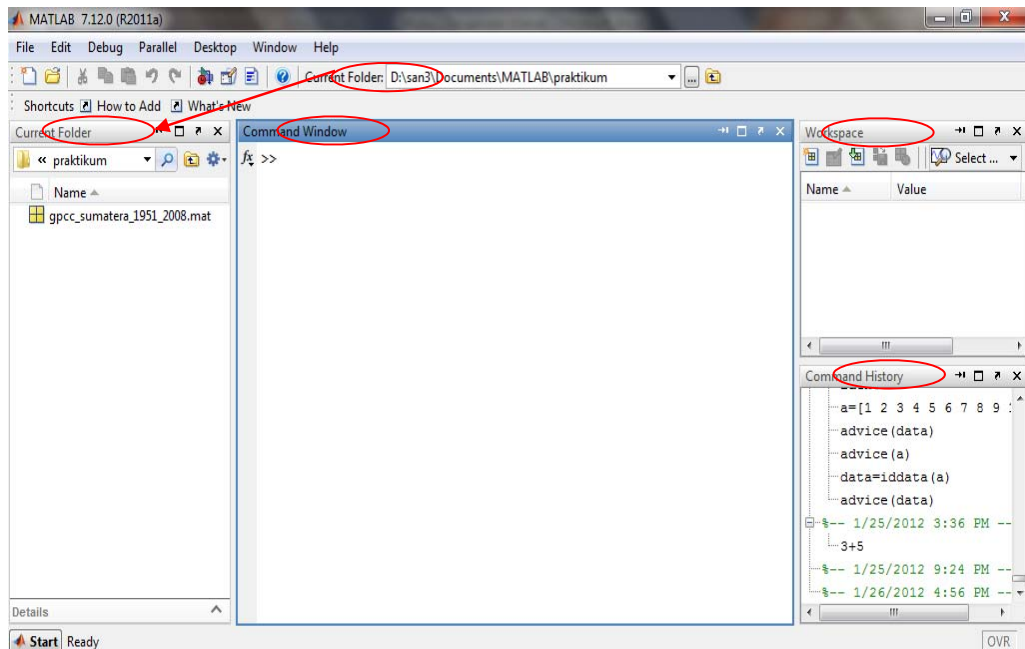
Ketika anda mulai membuka program Matlab, akan muncul desktop Matlab yang berisi tools (*Graphical user interface*) untuk mengatur file, variabel dan aplikasi yang berhubungan dengan Matlab.

Desktop matlab akan muncul seperti pada Gambar 10. Desktop tersebut terdiri dari berbagai macam *window* matlab yang letak dan tampilannya-nya bisa diatur sesuka hati pengguna. Untuk mengubah letak, bisa dilakukan dengan men-drag *window* yang ingin diubah. Sedangkan kemunculan tampilan dapat di atur pada menu → desktop lalu dicentang, atu di un-centang.



Gambar 7 Ilustrasi Desktop Matlab

Contoh fitur desktop matlab versi 7.12 :



Gambar 8 Desktop Matlab versi 7.12.0

Window yang muncul pada desktop matlab yaitu :

- *Current Folder*

Tempat direktori/folder kerja ditentukan. Pengguna dapat memilih letak direktori kerja yang akan dilakukan. Pemilihan direktori/folder dapat dilakukan di 2 tempat (*window*) yang fungsinya sama.

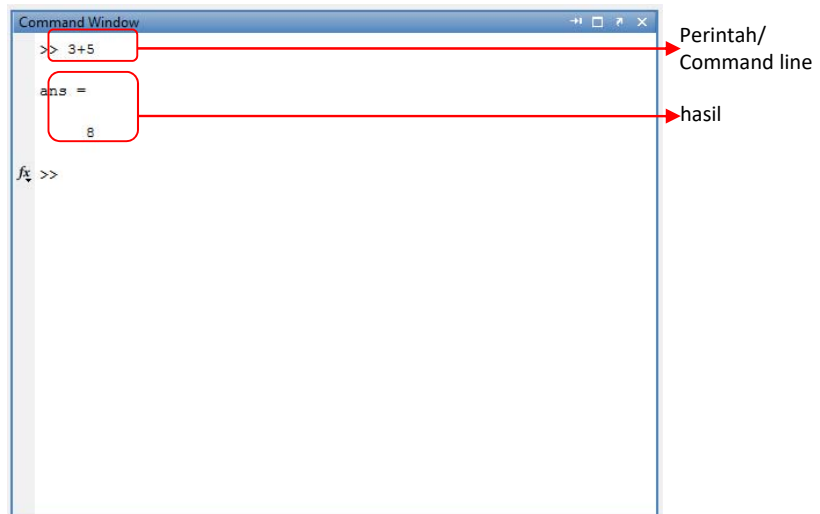
Pada contoh diatas pengguna menggunakan direktori/folder :

D:\san3\Documents\MATLAB\praktikum

Di dalam direktori/folder tersebut terdapat 1 file yaitu : gpcc_sumatera_1951_2008.mat muncul di *window* current folder sebelah kiri.

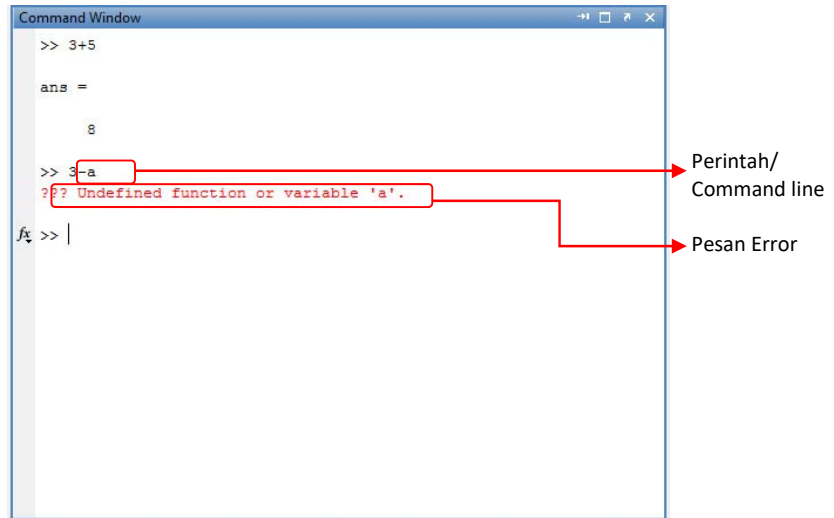
- *Command Window*

Tempat penulisan perintah/ *command line*. Bila perintah anda salah akan keluar pesan error. Prinsip dasar untuk memulai menggunakan Matlab, anda anggap Matlab adalah sebuah kalkulator. Untuk itu coba dengan mengerjakan operasi matematika sederhana layaknya sebuah kalkulator di *command window*.



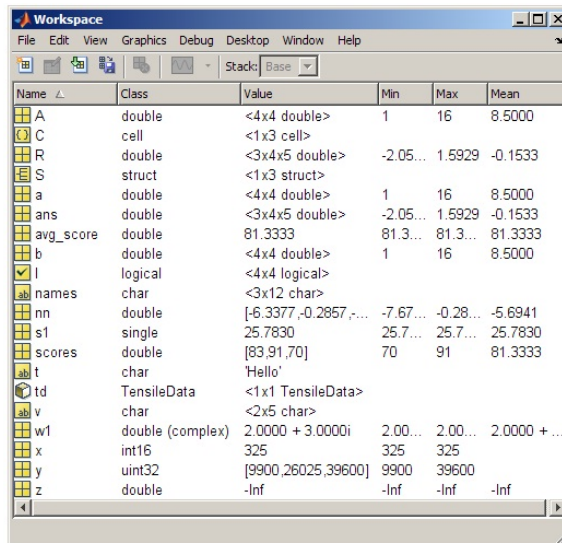
Gambar 9 Command Window

Pesan yang bertuliskan text berwarna merah muncul karena huruf 'a' yang dibuat untuk mengurangi angka 3 diatas belum didefinisikan (a adalah variable tertentu).



Gambar 10 Command Window Pesan Error

- Workspace
Window ini berisi keterangan atau informasi dari variable-variabel yang sedang dibuka atau dikerjakan. Tampilan lengkap akan seperti pada Gambar 14. Kolom-kolom pada workspace menerangkan nama variable, jenis kelas data yang tersimpan di dalam setiap variable, nilai data yang tersimpan, nilai minimum, maksimum dan rata-rata. Kolom-kolom informasi pada workspace dapat ditambah atau di kurangi dengan klik kanan pada judul kolom, lalu centang yang ingin dimunculkan.



Gambar 11 Workspace

Diktat Pelatihan SD (Statistical Downscaling)

Jika *window* ini tidak terbuka secara otomatis, maka data dibuka dengan memilih menu desktop→workspace, atau ketik 'workspace' di *command window*.

Selain itu, dapat juga dengan memilih menu File→Preferences→Toolbars untuk membuka box dialog.

Jika anda ingin membuka informasi data langsung di *command window*, dapat dengan mengetik 'who' pada *command window*.

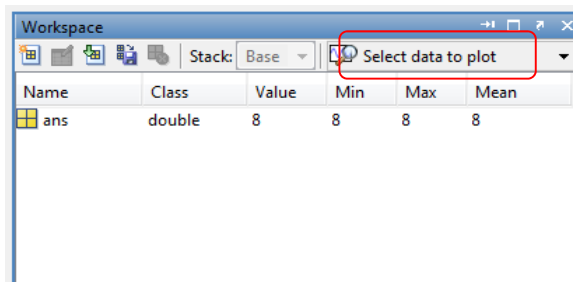
```
>> who
.AB Programs
Your variables are:

A      S      avg_score  names      scores      v      Y
C      a      b      nn      t      w1      z
R      ans     l      s1      td      x

>> whos
Name      Size      Bytes  Class      Attributes

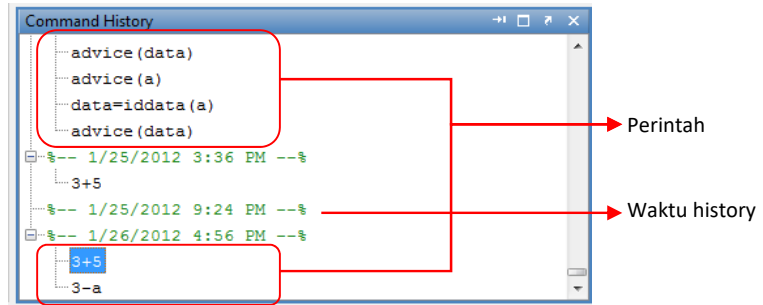
A          4x4          128  double
C          1x3          348  cell
R          3x4x5        480  double
S          1x3          826  struct
a          4x4          128  double
ans       3x4x5        480  double
avg_score  1x1           8  double
b          4x4          128  double
l          4x4           16  logical
names     3x12          72  char
nn        3x3           72  double
s1        1x1           4  single
scores    1x3           24  double
t         1x5           10  char
td        1x1          152  TensileData
v         2x5           20  char
w1        1x1           16  double      complex
x         1x1           2  int16
y         1x3           12  uint32
z         1x1           8  double
```

Pada workspace juga terdapat tools 'plot selector' yaitu menggambar data/variable tertentu yang dipilih. Plot selector adalah salah satu tool yang dapat digunakan untuk menggambar data dengan cepat pada Matlab.



Gambar 12 Workspace Plot Selector

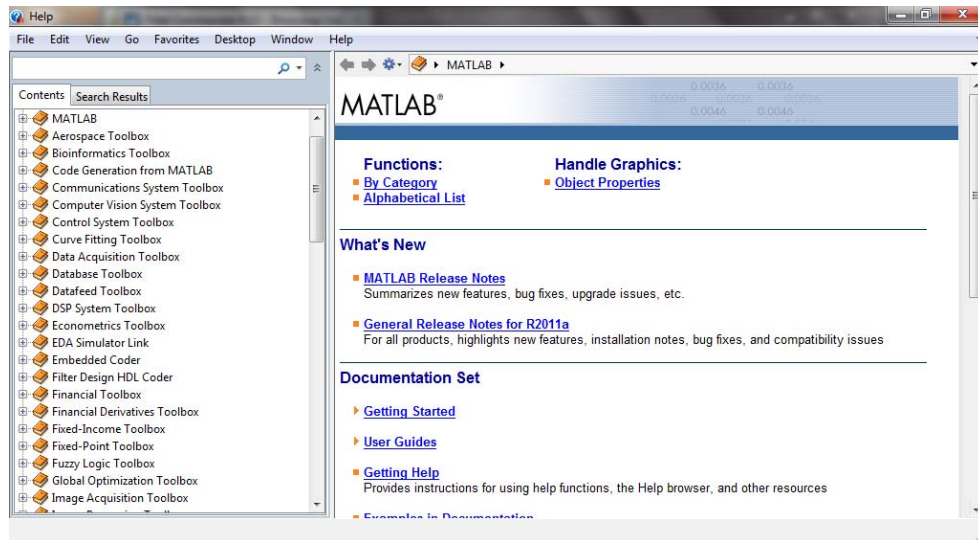
- Command History
Perintah yang telah ditulis di *command window*, akan tersimpan pada *command history*.



Gambar 13 Command History

Di command history anda dapat melihat statemen yang lalu dan mengkopi lalu mengeksekusi statemen yang dipilih.

Matlab juga menyediakan tombol 'Help' jika ingin melihat dan mencari dokumentasi dan demo untuk produk MathWork. Tombol ini telah disediakan pada Menu → Help. Menu ini sangat membantu jika anda ingin mencari fungsi, perintah atau keterangan tentang produk-produk MathWork.



Gambar 14 Window Help Matlab

4.2 Pemahaman Tentang Array dan Matrik

4.2.1 Dasar-dasar Array dan Matriks

Matlab merupakan operasi aritmatika per array secara elemen per elemen. Tujuan yang ingin dicapai dalam modul ini adalah untuk memahami cara membentuk array/matriks dan dapat melakukan operasi matematis array/matriks dalam Matlab.


Array adalah sekumpulan objek dengan atribut yang sama yang dapat diindekskan secara individual menggunakan beberapa teknik salah satunya dengan pengindeksan. Pada sistem penyimpanan data, array merupakan metode penyimpanan informasi dengan berbagai metode. Secara umum suatu array merupakan sejumlah item yang disusun dengan cara tertentu – misalkan, dalam list atau dalam tabel 3 dimensi. Untuk menyimpan beberapa nilai sekaligus dalam suatu variable dalam Matlab dibutuhkan variable array atau variable berindeks. Variabel array dapat digunakan untuk menyimpan banyak data yang sejenis baik untuk jenis data numeric maupun string.

Matriks dalam Matlab merupakan array yang terdiri dari angka-angka. Variabel dengan tipe data tunggal/matriks derajat nol (skalar) hanya dapat digunakan untuk menyimpan satu nilai. Operasi matriks dapat mencakup matriks 1 x 1 atau scalar, matriks dengan satu baris maupun satu kolom atau vektor, hingga operasi matriks berderajat lebih tinggi.


Matriks derajat nol – skalar

>> A = 5;	
-----------	---

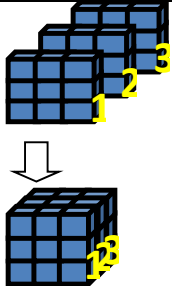
Matriks derajat 1 – deret, vektor, array → $A = [5 \ 7 \ 8]$

>> A(1) = 5; A(2) = 7; A(3) = 8;	
----------------------------------	---

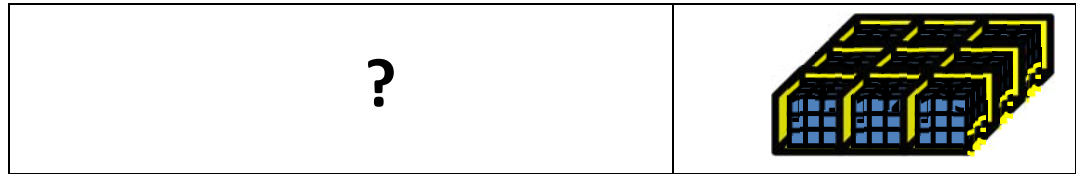
Matriks derajat 2 → $A = \begin{bmatrix} 5 & 7 & 8 \\ 3 & 9 & 2 \\ 4 & 6 & 1 \end{bmatrix}$

>> A(1,1) = 5; A(1,2) = 7; A(1,3) = 8; >> A(2,1) = 3; A(2,2) = 9; A(2,3) = 2; >> A(3,1) = 4; A(3,2) = 6; A(3,3) = 1;	
--	---

Matriks derajat 3

>> A(1,1,1) = 5; A(1,2,1) = 7; A(1,3,1) = 8; >> A(2,1,1) = 3; A(2,2,1) = 9; A(2,3,1) = 2; >> A(3,1,1) = 4; A(3,2,1) = 6; A(3,3,1) = 1; >> A(1,1,2) = 5; A(1,2,2) = 7; A(1,3,2) = 8; >> A(2,1,2) = 3; A(2,2,2) = 9; A(2,3,2) = 2; >> A(3,1,2) = 4; A(3,2,2) = 6; A(3,3,2) = 1; >> A(1,1,3) = 5; A(1,2,3) = 7; A(1,3,3) = 8; >> A(2,1,3) = 3; A(2,2,3) = 9; A(2,3,3) = 2; >> A(3,1,3) = 4; A(3,2,3) = 6; A(3,3,3) = 1;	
--	---

Matriks dengan derajat lebih tinggi



6.2.2 Operasi Array dan Matriks

Berikut ini contoh cara membuat array sederhana:

Diketahui nilai $x = n \times \pi$ dengan $n = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, \frac{4}{4}$. Bila $y = \sin(x)$ maka tentukan nilai y :

```
>> x = [0*pi 0.25*pi 0.5*pi 0.75*pi 1*pi]
x =
    0    0.7854    1.5708    2.3562    3.1416

>> y = sin(x)
y =
    0    0.7071    1.0000    0.7071    0.0000
```

Pengalamatan Array

Tiap nilai x dan y memiliki 5 elemen yang dipisahkan oleh kolom, Matlab memberikan identitas dengan kolom, yaitu array 1 kali 5. Untuk mengakses tiap elemennya digunakan indeks sebagai identitas tiap elemen. Elemen x pertama dapat diakses dengan $x(1)$, untuk x kedua $x(2)$ dan seterusnya.

```
>> x(1)
ans =
    0
>> x(2)
ans =
    0.7854
>> x(1:3)
ans =
    0    0.7854    1.5708
>> x(4:end)
ans =
    2.3562    3.1416
>> x(1:2:5)
ans =
    0    1.5708    3.1416
```

Orientasi Array

Untuk membuat terdapat dua macam orientasi yaitu disebut vektor baris bila array berorientasi pada baris, terdiri dari satu baris dengan beberapa kolom dan disebut vektor kolom bila array terdiri dari satu kolom dengan beberapa baris.

```
>> A = 1:3
A =
     1     2     3
>> B = [1;2;3]
B =
     1
     2
     3
>> C = A'
C =
     1
     2
     3
```

Membuat Array

Pada contoh sebelumnya, array dibuat dengan meng-input nilai yang diketahui satu per satu. Berikut ini dapat digunakan metode yang lebih mudah dalam membuat array menggunakan notasi kolom.

```
>> x = (0:0.25:1)*pi
x =
     0     0.7854     1.5708     2.3562     3.1416

Linspace(nilai_awal, nilai_akhir, jumlah_elemen)
>> x = linspace(0,pi,5)
x =
     0     0.7854     1.5708     2.3562     3.1416
```

Ukuran Array

Tabel 2 Syntaq dan keterangan ukuran array

Syntaq	Keterangan
whos	Menampilkan variabel yang digunakan beserta ukurannya
x = size(A)	Menghasilkan vektor baris x dengan elemen pertama jumlah baris dan elemen kedua jumlah kolom A
[x,y] = size(A)	Menghasilkan dua nilai scalar x dan y dimana x adalah jumlah baris dan y adalah jumlah kolom A
x = size(A,1)	Menghasilkan scalar variabel x yang menunjukkan jumlah baris A
y = size(A,2)	Menghasilkan scalar variabel y yang menunjukkan jumlah kolom A
n = length(A)	Menghasilkan scalar variabel n yang menunjukkan panjang vektor A

Contoh 1.


```
>> whos
  Name      Size      Bytes  Class      Attributes
  A         1x3         24    double
  B         3x1         24    double
>> x = size(A)
x =
     1     3
>> [x,y] = size(A)
x =
     1
y =
     3
>> x = size(A,1)
x =
     1
>> y = size(A,2)
y =
     3
>> n = length(A)
n =
     3
```

6.2.3 Operasi Elemen per Elemen

Operasi aritmatika pada suatu array dilakukan dengan cara elemen per elemen. Hal ini berarti penjumlahan dan pengurangan seperti operasi pada matriks, namun operasi perkalian/pembagiannya berbeda. Untuk operasi array, Matlab menggunakan operator titik atau titik decimal sebagai bagian dari notasi untuk operasi perkalian array. Berikut ini adalah list operator yang digunakan:

Tabel 3 Notasi yang digunakan dalam operasi elemen

+	Addition
-	Substraction
.*	Element-by-element multiplication
./	Element-by-element division
.\	Element-by-element left division
.^	Element-by-element power
.'	Unconjugate array transpose

Contoh 1. Penjumlahan/pengurangan array-skalar dan array-array, penjumlahan array-array

Diketahui suatu matriks A dan B:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \text{ dan } B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Tentukan:

$$C = A - (5 \times B)$$
$$D = A + B$$

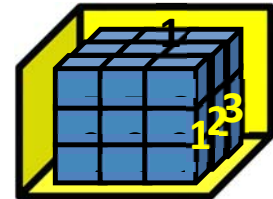
```
>> A = [1,2,3;4,5,6;7,8,9];  
>> B = [1,0,0;0,1,0;0,0,1];  
>> C = A-5*B  
>> D = A+B
```

Contoh 2. Perkalian elemen array-array

Syarat operasi elemen array –array dalam Matlab adalah panjang elemen dalam satu dimensi ke semua dimensi lain harus sama. Misalkan, panjang kolom dalam setiap baris harus sama, dan panjang baris dalam setiap kolom harus sama.

Diketahui suatu matriks A dan B:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \text{ dan } B = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix}$$



Tentukan:

$$C = A \times B$$
$$D = C \div A$$

```
>> A = [1,2,3;4,5,6];  
>> B = [2,2,2;2,2,2];  
>> C = A.*B  
>> D = C./B
```

Penanda khusus yang digunakan: , [] [;] ; () (:)

Operasi Matrik

Berbeda dengan perkalian elemen per elemen array, perkalian matriks dalam Matlab menggunakan operator asterik (*). Syarat operasi perkalian matriks apabila:

X memiliki dimensi $A \times B$ dan

Y memiliki dimensi $C \times D$ maka

Syaratnya $B = C$ dan dimensi matriks yang dihasilkan adalah $A \times D$.

Contoh 1. Perkalian Matrik, Determinan, dan Invers

Diketahui matrik A dan vektor B dengan nilai sbb:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 5 \end{bmatrix} \text{ dan } B = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Tentukan:

Nilai perkalian matriks dari A dikalikan B

Determinan dari matriks A

Invers dari matriks A

```
>> A = [1,2,3;4,5,6;7,8,5];  
>> B = [1;2;3];  
>> C = A*B;  
>> det(A)  
>> inv(A)
```

Contoh 2. Penyelesaian persamaan linear

Tentukan nilai x, y dan z dari ketiga persamaan linier berikut:

$$x + 2y + 3z = 366$$

$$4x + 5y + 6z = 804$$

$$7x + 8y = 351$$

Persamaan linier di atas dapat diubah menjadi bentuk matriks sbb:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 366 \\ 804 \\ 351 \end{bmatrix}$$

Misalkan:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}, B = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, C = \begin{bmatrix} 366 \\ 804 \\ 351 \end{bmatrix} \text{ maka } A \cdot B = C$$

Nilai B dapat dicari salah satunya dengan metode invers sbb:

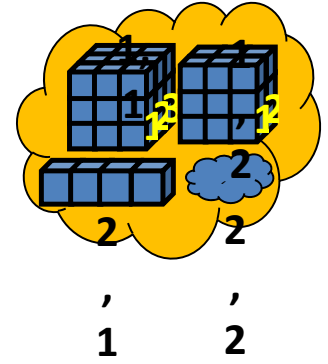
$$B = A^{-1} \cdot C$$

```
>> A = [1,2,3;4,5,6;7,8,0];  
>> C = [366;804;351];  
>> B = inv(A)*C  
B =  
    25.0000  
    22.0000  
    99.0000  
>> x = B(1);  
>> y = B(2);  
>> z = B(3);
```

Cell Array → array dalam array

Cell Array dalam Matlab merupakan array multidimensi yang memiliki elemen-elemen salinan dari array yang lain. Suatu cell array kosong dapat dibuat dengan cell function.

Namun cell array lebih sering dibuat dengan memasukkan berbagai macam kumpulan dari berbagai hal dalam tanda kurung kurawal { }. Tanda kurung kurawal juga digunakan dengan subscripts untuk mengakses isi dari berbagai sel.



```
>> D = {A, B, [1 4 6 9 10], 'test', {'coba'}}
D =
    [3x1 double]    [1x3 double]    [1x5 double]    'test'    {1x1
cell}
>> D(1) → ?
>> D(2) → ?
>> D(3) → ?
>> D(4) → ?
>> D(5) → ?
>> E = {A; B; 'test'; {'coba'}}
```

Cell Array boleh berbeda-beda ukurannya, bahkan boleh terdiri dari campuran antara numerik dan character.

Penanda khusus yang digunakan: { }

Struct array → seperti cell array namun indeks yang digunakan bukan numerik.

Structure adalah obyek Matlab dengan *field* atau kontainer data. *Field* dari suatu struktur dapat memuat segala jenis tipe data. Perbedaan antara cell array dengan struktur adalah *field* dari *structure* dapat diakses dengan namanya, bukan dengan indeks, serta tidak terdapat batasan tambahan pada jumlah atau konfigurasi *field* dari suatu *structure*. Seperti halnya pada *cell array* suatu *structure* dapat memuat *structure* lain dan dimasukkan pada array atau *cell array*.

Membuat Structure

Notasi titik (.) digunakan untuk mengakses *field* dalam *structure*. Untuk membuat *structure* dapat dilakukan dengan mengisi data ke dalam *field*.

Contoh:

Berikut ini adalah contoh membuat *struct* data:

```
>> data.variabel = 'Annual Rainfall';
>> data.periode = [2000 2005];
>> data.value = [1000; 1200; 1500; 1100; 900];
>> data.unit = 'mm';
>> data.wilayah = 'Stasiun Cemara Bandung';
```

Selanjutnya, buat data kedua berupa data temperatur.

```
>> data(2).variabel = 'Mean Annual Temperature';  
>> data(2).periode = [2001 2009];  
>> data(2).value = [27; 25; 29; 27; 26; 25; 27; 28; 25];  
>> data(2).unit = 'deg.Celcius';  
>> data(2).wilayah = 'Stasiun Cemara Bandung';
```

```
data =  
1x2 struct array with fields:  
    variabel  
    periode  
    value  
    unit  
    wilayah
```

Mengakses *field structure*

```
>> data.variabel  
ans =  
Annual Rainfall  
ans =  
Mean Annual Temperature  
  
>> data(2)  
ans =  
    variabel: 'Mean Annual Temperature'  
    periode: [2001 2009]  
    value: [9x1 double]  
    unit: 'deg.Celcius'  
    wilayah: 'Stasiun Cemara Bandung'  
  
>> data(2).value  
>> ch = data(2).value  
>> T = fieldnames(data)
```

Menghapus *field structure*:

```
>> data = rmfield(data,'wilayah')  
data =  
1x2 struct array with fields:  
    variabel  
    periode  
    value  
    unit
```

Penanda khusus yang digunakan: .

Operasi-operasi matriks khusus

```

>> A = ones (3,3)  →  A =
                        1    1    1
                        1    1    1
                        1    1    1

>> A = zeros(3)    →  A =
                        0    0    0
                        0    0    0
                        0    0    0

>> B = zeros (3,4) →  B =
                        0    0    0    0
                        0    0    0    0
                        0    0    0    0

>> C = eye (3,3)   →  C =
                        1    0    0
                        0    1    0
                        0    0    1

>> D = rand (3,4)  →  D = 0.7547    0.6551    0.4984    0.5853
                        0.2760    0.1626    0.9597    0.2238
                        0.6797    0.1190    0.3404    0.7513

>> E = D'          →  E = 0.7547    0.2760    0.6797
                        0.6551    0.1626    0.1190
                        0.4984    0.9597    0.3404
                        0.5853    0.2238    0.7513

```

4.3 Fungsi Dasar

Matlab mempunyai berbagai fungsi matematika umum yang biasa di gunakan dalam matematika. Sebagian besar fungsi tersebut hampir sama dengan bila anda menuliskannya secara matematis.

Tabel 4 Fungsi Dasar Matematika

Fungsi Matematika Dasar	Deskripsi
Abs	nilai absolute atau amplitudo bilangan kompleks
Angle	sudut fasa
Sqrt	akar kuadrat
Real	bagian real dari bilangan kompleks
Imag	bagian imaginer dari bilangan kompleks
Conj	konjugat bilangan kompleks
Round	pembulatan ke bilangan bulat terdekat.
Fix	pembulatan ke arah nol
Floor	pembulatan ke arah $-\infty$
Ceil	pembulatan ke arah ∞
Rem	sisia
Exp	exponensial berbasis bilangan e

Log	logaritma murni
Log10	logaritma basis 10

Tabel 5 Fungsi Trigonometri

Fungsi Trigonometri	Deskripsi
in, asin, sinh, asinh cos, acos, cosh, acosh tan, atan, tanh, atanh cot, acot, coth, acoth sec, asec, sech, asech csc, acsc, csh, acsh	sinus, anti sinus, hiperbolik sinus, hiperbolik anti sinus cosines, anti cosines, hiperbolik cosines, hiperbolik anti cosines tangent, anti tangent, tangent hiperbolik, anti tangent hiperbolik cotangent, anti cotangent, cotangent hiperbolik, anti cotangent hiperbolik. secan, antisechan, secan hiperbolik, anti secan hiperbolik cosecant, anti cosecant, cosecant hiperbolik, anti cosecant hiperbolik.

Contoh :

```
>> pi
ans =
3.1416
>> cos(pi)
ans =
-1
```

Misalkan bilangan A adalah nilai 1-10, carilah nilai rata-ratanya, nilai maksimumnya dan nilai minimumnya.

```
>> A=[1 2 3 4 5 6 7 8 9 10];
>> mean(A)
ans =
5.5000
>>max (A)
ans =
10
>>min (A)
ans = 1
```

Operator Dasar Aritmatika

Ekspresi yang sering digunakan dalam operasi matematika dasar dalam matlab meliputi prosedur berikut :

Tabel 6 Operator Dasar Aritmatika

Operator	Deskripsi
+	Penjumlahan
-	Pengurangan
*	Pengalian

/	Pembagian
\	Pembagian dari kiri
^	Pangkat
'	Transpose
()	Printah pengevaluasian yang spesifik

Operator Logika

Ekspresi yang sering digunakan dalam untuk operator logika dalam matlab meliputi prosedur berikut:

Tabel 7 Operator Logika

Operator	Deskripsi
<	kurang dari
<=	kurang dari sama dengan
>	lebih dari
>=	lebih dari sama dengan
==	sama dengan
~=	tidak sama dengan
	or
&	and
~	not

4.4 Input/Output Dasar

Untuk menyimpan dan memanggil data dari file pilih File → Save Workspace As ...

Untuk memanggil data digunakan pilihan Load WorkSpace As atau Open pada menu file.

Sedangkan untuk mengimport data , pilih file→ Import Data .. .

Matlab juga menyediakan dua perintah --save dan load -- yang jauh lebih fleksibel. Perintah save untuk menyimpan satu atau lebih variabel dalam file format yang sesuai dengan pilihan anda.

Contoh :


```
>>clear all
>>x=1:10;y=10:10:10:100;      % membuat array baru
```

Disini dicontohkan dua buah array yaitu array x dan y. Ada beberapa cara menyimpan data dalam matlab, seperti :

```
>>save
Saving to: Matlab.mat
menyimpan semua variabel Matlab dalam format biner di file Matlab.mat
```

```
>>save data
menyimpan semua variabel Matlab dalam format biner di file data.mat
```

```
>>save data_x x
menyimpan variabel x dalam format biner di file data_x.mat
```

```
>>save data_xy.dat x -ascii
menyimpan variabel x dalam format biner di file data_xy.dat dalam format ascii. untuk membuka data digunakan perintah load, contoh;
>>load data_x.mat
```

```
>>save data_x.txt x
Menyimpan variable x dalam format ascii dengan file ekstensi .txt
```

File-file tersebut akan tersimpan di folder/direktori kerja yang sedang dibuka. Akan muncul juga di window *current folder*.

4.5 Visualisasi Matlab

Dalam bab ini akan dibahas berbagai fasilitas grafis Matlab secara lebih jelas. Matlab menyediakan berbagai fungsi untuk menampilkan data secara dua dimensi maupun tiga dimensi.

1. Perintah-perintah Dasar
 - a. Plot
Perintah plot sering digunakan untuk menggambarkan grafik dua dimensi. Perintah plot menggambarkan data dalam array pada sumbu yang bersesuaian dan menghubungkan titik-titik tersebut dengan garis lurus.

Syntax :

- plot(Y)

Memplot kolom variabel Y terhadap indexnya.

Contoh :

```
y=[1 3 4 8 5 7 2 0];
```

```
plot(y)
```

- plot(X1,Y1,...)

Memplot semua garis yang didefinisikan oleh variabel Xn terhadap Yn.

Contoh visualisasi dari $y = \frac{\sin x}{x}$ dengan $x = \left[\frac{\pi}{100}, 10\pi \right]$:

```
x=pi/100:pi/100:10*pi;
```

```
y=sin(x)./x;
```

```
plot(x,y)
```

- plot(X1,Y1,LineStyle,...)

Memplot semua garis yang didefinisikan oleh variabel Xn terhadap Yn. LineSpec merupakan spesifikasi garis (tipe, simbol marker, dan warna garis).

Tabel 8 Simbol Garis

Specifier	Line Style
-	Solid line (default)
--	Dashed line
:	Dotted line
-.	Dash-dot line

Tabel 9 Simbol Warna

Specifier	Color
r	Merah
g	Hijau
b	Biru
c	Cyan
m	Magenta
y	Kuning
k	Hitam
w	Putih

Tabel 10 Tanda (Marker)

Specifier	Marker Type
+	Plus sign
o	Circle
*	Asterisk
.	Point
x	Cross
'square' or s	Square
'diamond' or d	Diamond
^	Upward-pointing triangle
v	Downward-pointing triangle
>	Right-pointing triangle

<	Left-pointing triangle
'pentagram' or p	Five-pointed star (pentagram)
'hexagram' or h	Six-pointed star (hexagram)

Contoh untuk nilai x dan y yang sama :

```
plot(x,y,'+r')
```

- `plot(...,'PropertyName',PropertyValue,...)`

Menentukan properti ke nilai properti tertentu. Nama-nama properti dan nilai-nilainya secara lengkap dapat dilihat di dalam dokumentasi Matlab dengan judul *Lineseries Properties*.

Contoh untuk nilai x dan y yang sama :

```
plot(x,y,'-r', 'LineWidth',2, 'MarkerEdgeColor','k',  
'MarkerFaceColor','g', 'MarkerSize',10))
```

- `plot(axes_handle,...)`

- `h = plot(...)`

- `hlines = plot('v6',...)`

b. Scatter

Syntax :

- `scatter(X,Y,S,C)`

- `scatter(X,Y)`

- `scatter(X,Y,S)`

- `scatter(...,markertype)`

- `scatter(...,'filled')`

- `scatter(...,'PropertyName',propertyvalue)`

- `scatter(axes_handles,...)`

- `h = scatter(...)`

- `hpatch = scatter('v6',...)`

X dan Y : menentukan posisi marker

S : menentukan ukuran marker

C : menentukan warna marker

Markertype : tipe marker, sama seperti marker specifier dalam LineSpec

Filled : jika menginginkan marker terisi

PropertyName dan *propertyvalue* dapat dilihat secara lengkap dalam dokumentasi Matlab dengan judul *Scattergroup Properties*.

Contoh dengan nilai x dan y yang sama :

```
scatter(x,y,5,'g')
```

2. Handle Graphics dan Properti Gambar

a. Handle

Handle Grapics bisa dikatakan referensi dari data. Handle Grapics dapat disimpan ketika kita membuat objek tertentu, atau dimuat dalam suatu variabel yang menyimpan informasi dari objek tersebut, ataupun diperoleh dari beberapa fungsi dalam Matlab seperti `findobj` atau `gca`.

Diktat Pelatihan SD (Statistical Downscaling)

Jika kita memasukkan handle ke dalam suatu variabel, variabel ini nantinya akan digunakan sebagai inputan perintah *set* yang digunakan untuk memanipulasi gambar.

Dalam pembahasan sebelumnya, terdapat syntax `h=plot(...)` dan `h=scatter(...)`. `h` di sini merupakan variabel yang mewakili handle.

b. Properti Gambar

Dalam Matlab, terdapat beberapa properti yang dapat digunakan untuk memanipulasi gambar agar sesuai dengan apa yang kita inginkan. Properti ini dikelompokkan dalam line property, axes property, dan figure property.

Beberapa line properties yang sering digunakan:

Tabel 11 Contoh Properti Garis

Nama Properti	Deskripsi	Contoh Nilai Properti
LineStyle	Bentuk garis	'-', ':', '-.'
Marker	Simbol marker	'+', 'o', '*'. '.', 'x', dll
MarkerSize	Ukuran simbol marker	1, 2, 3, 4, ... (dalam point, 1 point=1/72 inci)
XData, YData, ZData	Vektor dari koordinat x, y, dan z yang mendefinisikan garis sumbunya.	

Properti selengkapnya dapat dilihat di dokumentasi Matlab dengan judul *Line Properties*.

Beberapa axes properties yang sering digunakan:

Tabel 12 Properti Axis

Nama Properti	Deskripsi	Contoh Nilai Properti
FontName	Jenis huruf untuk label axes	Courier, Arial
FontSize	Ukuran huruf untuk label axes dan judul gambar	12, 13, 14,
LineWidth	Ketebalan garis axis	0.5, 1, 1.5, ...
XLabel, YLabel, Zlabel	Handel dari label axis	Bilangan acak
XTick, YTick, ZTick	Ukuran/lebar spasi antar marker data	10, 12, 24, dll
XTickLabel, YTickLabel, ZTickLabel	Label dari sumbu x, y, dan z	{'One';'Two';'Three';'Four'}, atau angka

Properti selengkapnya dapat dilihat di dokumentasi Matlab dengan judul *Axes Properties*.

Beberapa figure properties yang sering digunakan:

Tabel 13 Properti Gambar

Nama Properti	Deskripsi	Contoh Nilai Properti
FileName	Nama gambar	Dalam string
Position	Posisi gambar	left, bottom, width, height

Name	Judul jendela gambar	Dalam string

Properti selengkapnya dapat dilihat di dokumentasi Matlab dengan judul *Figure Properties*.

Selain itu, untuk memberi label pada sumbu grafik, bisa digunakan fungsi *xlabel*, *ylabel*, dan *zlabel*, juga *title* untuk memberi judul gambar. Syntax :

```
xlabel/ylabel/zlabel/title('string')  
Contoh dengan menggunakan nilai x dan y yang sama :  
h=plot(x,y)  
set(h,'LineWidth',2,'LineStyle',':', 'Color','y')  
xlabel('Sumbu x')  
ylabel('Sumbu y')  
title('Latihan')
```

Properti dalam fungsi set bisa merupakan line property, axes property, atau figure property.

3. Subplot

Fungsi subplot digunakan untuk menampilkan beberapa gambar dalam satu jendela.

Syntax :

- `h = subplot(m,n,p) or subplot(mnp)`
- `subplot(m,n,p,'replace')`
- `subplot(m,n,P)`
- `subplot(h)`
- `subplot('Position',[left bottom width height])`
- `subplot(..., prop1, value1, prop2, value2, ...)`
- `h = subplot(...)`
- `subplot(m,n,p,'v6')`

`m` dan `n` : nilai matriks, untuk membagi satu jendela menjadi matriks `m x n` area grafik.

`i` : indeks penomoran gambar. Subplot dinomori dari kiri ke kanan dimulai dari baris teratas.

`replace` : jika di dalam jendela sudah ada gambar tertentu, gambar tersebut akan dihapus dan diganti dengan gambar baru.

`P` : dalam vektor, menentukan suatu posisi gambar yang diliputi oleh semua posisi subplot menjadi satu gambar.

`h` : handel

Contoh :

```
x1=pi/100:pi/100:5*pi;  
x2= pi/100:pi/100:10*pi;  
x3= pi/100:pi/100:15*pi;  
y1=sin(x1)./x1;  
y2=sin(x2)./x2;  
y3=sin(x3)./x3;
```

```
subplot(2,2,1);plot(x1,y1)
subplot(2,2,2);plot(x2,y2)
subplot(2,2,[3 4]);plot(x3,y3)
```

BAB 7

BAB 5 DASAR-DASAR *SCRIPTING* PADA MATLAB

Bahasa *Script* adalah bahasa pemrograman yang memungkinkan kita melakukan kontrol terhadap satu aplikasi atau lebih. *Script* adalah semacam program yang terdiri dari sebuah set instruksi-instruksi untuk sebuah aplikasi atau utilitas program. Dalam Matlab file program/*script* disimpan dalam bentuk ekstensi .m atau biasa disebut “M-file”.

Buat lah m-file sederhana untuk menghitung konversi suhu. Untuk membuat m-file baru, pilih tab *File* → *New* → *script*.

```
%Contoh file script sederhana
%Program konversi suhu sederhana

c=input('Masukkan suhu Celcius yang akan dikonversi: ');

f=(9/5)*(c+32);

disp(['Hasil: ',num2str(f)]);
```

Setelah selesai simpan *script* anda kemudian coba eksekusi. *Script* anda akan ditampilkan pada *command window*.

5.1 Kontrol Program

Kontrol program sangat berguna karena memungkinkan proses komputasi sebelumnya dapat mempengaruhi proses komputasi berikutnya. Dalam Matlab kita dapat mengatur jalannya eksekusi perintah-perintah dalam program dengan didasarkan pada struktur pengambilan keputusan. Materi ini mungkin sedikit agak rumit bagi anda yang belum pernah menggunakan kontrol program sebelumnya.

Matlab menyediakan empat struktur pengambilan keputusan atau struktur kontrol program, keempatnya adalah: *loop for*, *loop while*, konstruksi *if-else-end*, dan konstruksi *switch-case*. Konstruksi tersebut seringkali melibatkan banyak perintah Matlab, oleh karena itu konstruksi lebih sering terdapat dalam M-file daripada diberikan langsung dari *prompt* Matlab.

5.1.1 Loop for

Loop for memungkinkan sekelompok perintah diulang sebanyak suatu jumlah yang tetap. Bentuk umum *loop for* adalah:

```
for x= array
    Perintah
end
```

Contoh:

```
%Contoh file script sederhana
%Program konversi suhu menggunakan loop for

for c=(10:2.5:25);
    f=(9/5)*(c+32);
    disp([num2str(c) ' degC = ',num2str(f) ' degF']);
end;
```

5.1.2 Loop while

Loop for mengulang perintah sebanyak jumlah yang telah kita tentukan, sedangkan *loop while* akan mengulang perintah tak terbatas atau hingga kondisi yang telah kita tentukan. Bentuk umum *loop while* adalah:

```
while ekspresi
    Perintah
end
```

Contoh :

```
%Contoh file script sederhana
%Program menghitung faktorial

c=input('Masukkan bilangan bulat positif: ');
i=1;
fakt=1;
while i<=c;
    fakt=fakt*i;
end;
```

```
    i=i+1;  
end;  
disp([num2str(c) '! adalah: ',num2str(fakt)]);
```

5.1.3 Switch-case

Bila suatu perintah dikerjakan dengan didasarkan pada suatu tes yang berulang dengan argumen yang sama, konstruksi switch-else akan lebih tepat digunakan. Konstruksi ini berbentuk:

```
switch ekspresi  
    Case tes ekspresi 1  
        Deret perintah 1  
    Case tes ekspresi 2  
        Deret perintah 2  
    otherwise  
        Deret perintah 3  
end
```

Contoh:

```
bilangan=input('Masukkan angka: ');  
x=mod(bilangan,2);  
switch x  
case 1  
    disp(['bilangan ', num2str(bilangan), ' adalah bilangan ganjil'])  
case 0  
    disp(['bilangan ', num2str(bilangan), ' adalah bilangan genap'])  
otherwise  
    disp('Bilangan ini tidak mungkin ada')  
end
```

5.1.4 If-else-end

Seringkali sederetan perintah harus dikerjakan dengan didasarkan pada hasil tes rasional. Dalam bahasa pemrograman, logical ini dikerjakan dengan variasi konstruksi if-else-end. Bentuk paling sederhana konstruksi if-else-end adalah:

```
if ekspresi  
    Perintah  
end
```

Untuk kondisi yang membutuhkan lebih dari satu kondisi, konstruksi if-else-end berbentuk:


```
if ekspresi 1
    perintah dikerjakan jika ekspresi 1 benar
elseif ekspresi 2
    perintah dikerjakan jika ekspresi 2 benar
elseif ekspresi 3
    perintah dikerjakan jika ekspresi 3 benar
...
else
    perintah dikerjakan jika tidak ada ekspresi yang benar
end
```

Contoh:

```
x=input('Masukkan angka: ')
if x<0
    disp('Bilangan negatif');
elseif x>0
    disp('Bilangan positif');
else
    disp('Nol');
end
```

5.2 Fungsi M-File

Fungsi-fungsi dalam Matlab seperti `inv`, `abs`, `sqrt`, dan `exp` menerima variable yang anda berikan, menghitung hasil berdasarkan input dan mengembalikan hasil akhirnya pada anda sendiri. Bila diibaratkan fungsi adalah sebuah kotak hitam dimana yang kita lihat hanyalah apa yang masuk dan apa yang keluar.

Sifat-sifat demikian membuat fungsi dapat menjadi suatu alat yang baik untuk mengevaluasi perintah-perintah yang menggabungkan fungsi-fungsi matematika atau deretan perintah yang sering digunakan untuk memecahkan suatu masalah besar.

5.2.1 Aturan dan sifat

Fungsi m-file harus mengikuti beberapa aturan dan sejumlah sifat penting. Aturan dan sifat tersebut meliputi:

- Nama fungsi dan nama file harus identik misalnya anda membuat fungsi dengan nama pangkat maka anda member nama m-file anda dengan pangkat juga.
- Baris komentar sampai dengan baris bukan komentar yang pertama adalah teks help yang ditampilkan jika anda meminta help dari fungsi yang anda buat.
- Setiap fungsi mempunyai ruang kerjanya sendiri yang berbeda dengan ruang kerja Matlab. Satu-satunya hubungan antara ruang kerja matlab dengan variabel-variabel dalam fungsi adalah variabel-variabel input dan output fungsi. Jika suatu fungsi

mengubah nilai dalam suatu variabel input, perubahan itu hanya tampak dalam fungsi dan tidak mempengaruhi variabel yang ada dalam ruang kerja matlab.

- Jumlah dari argument input dan output yang digunakan jika suatu fungsi dipanggil hanya ada dalam fungsi tersebut. Fungsi ruang kerja memuat jumlah argument input. Fungsi kerja nargout memuat jumlah argument output. Dalam praktek, variabel-variabel nargout dan nargin biasanya digunakan untuk mengatur variabel input standard an menentukan variabel output yang diperlukan.

Syntax untuk membuat fungsi adalah sebagai berikut:

function y = nama_fungsi (x)

y adalah keluaran fungsi, keluaran ini dapat satu variabel atau lebih dari satu variabel, jika keluaran lebih dari satu variabel maka mempunyai bentuk sebagai berikut:

function [y,z,a,b] = nama_fungsi (x)

x adalah masukan dari fungsi, masukan ini dapat satu variabel atau lebih dari satu variabel, jika masukan lebih dari satu variabel maka mempunyai bentuk sebagai berikut:

function y = nama_fungsi (a,b,c,d)

Contoh:

```
%function y = humps (x)
%masukkan nilai x akan didapat nilai y
% y = 1./((x-0.3).^2+0.01)+1./((x-0.9).^2+0.04)-6

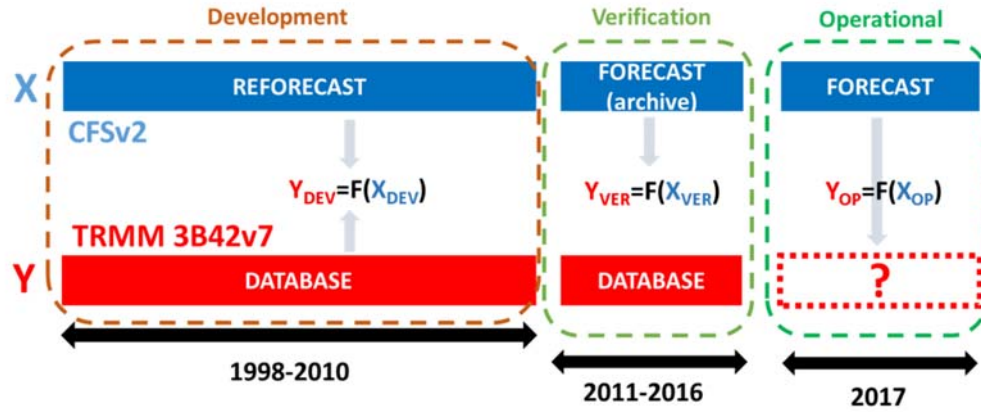
function y = humps (x)
y = 1./((x-0.3).^2+0.001)+1./((x-0.9).^2+0.04)-6;
```

Beri nama script sesuai dengan nama fungsi. Kemudian eksekusi.

BAB 6

BAB 6 TEKNIK SD (STATISTICAL DOWNSCALLING)

Di sini kita akan melakukan eksperimen statistical downscaling keluaran model global. Sebagai host model, akan digunakan model NCEP-CFSv2 (National Centres for Environmental Prediction-Climate Forecast System Version 2). Sementara sebagai pengganti database observasi lokal, akan digunakan data TRMM (Tropical Rainfall Measuring Mission) Precipitation Analysis (TMPA) versi ke-7. Data akan dibagi menjadi dua periode: periode *development (training)* dan periode *verification*. Periode development merupakan periode membentuk persamaan fungsi transfer atau model statistik yang menjelaskan hubungan antara model global dan kondisi lokal. Sementara itu, periode verification merupakan periode dimana model statistik tersebut diimplementasikan untuk men-downscale keluaran model global. Ilustrasi mengenai skema statistical downscaling ini dapat dilihat pada Gambar 18. Metode statistical downscaling yang akan dibahas adalah metode Statistical Bias-Correction (Quantile Based) dan Multi-Window Constructed Analogue.

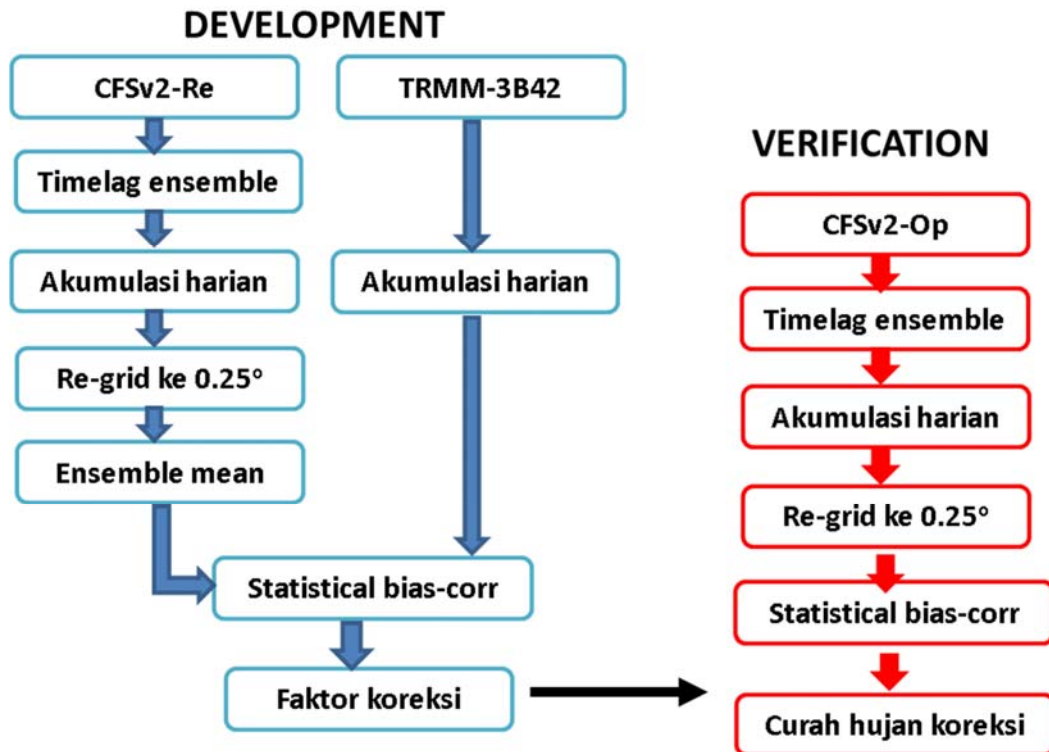


Gambar 15 Skema umum statistical downscaling dari keluaran model CFSv2. $Y=F(X)$ menyatakan fungsi transfer/ model statistik yang menjelaskan hubungan antara prediktor X (keluaran model global) dan prediktan Y (kondisi lokal)

6.1 Statistical Bias Correction dan Constructed Analogue

6.1.1 Data dan Flowchart

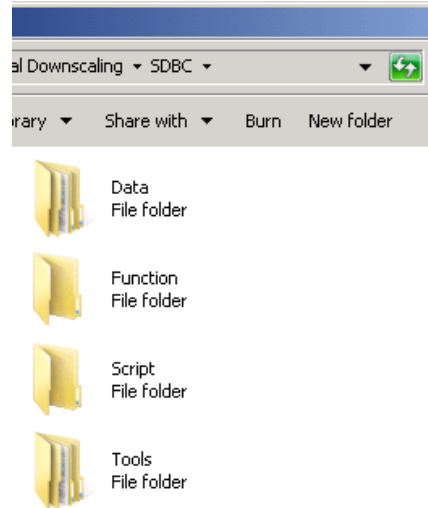
Data yang digunakan dalam eksperimen ini adalah curah hujan keluaran CFSv2 (sebagai prediktor) dan curah hujan lokal yang diwakili oleh TMPA (sebagai prediktan). Data CFSv2 sendiri terbagi atas dua periode, CFS-Reforecast yang digunakan di periode development/training, dan data CFSv2-Operational yang digunakan di periode Implementation/verification. Diagram alir eksperimen statistical downscaling dengan metode Statistical Bias Correction ditunjukkan dalam Gambar 19.



Gambar 16 Flowchart metode statistical bias-correction

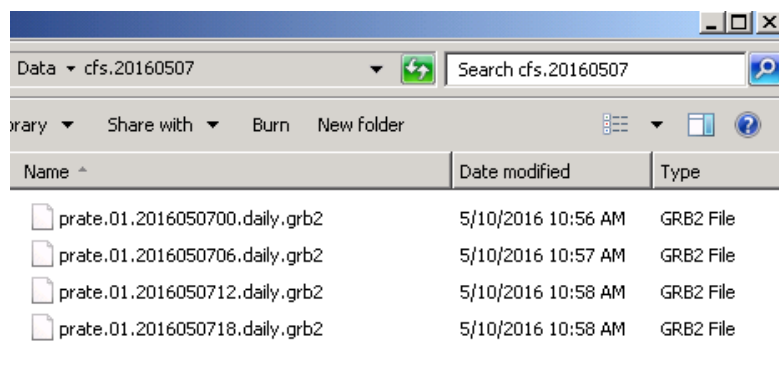
6.1.2 Struktur Direktori, Tools, Dan Script

Semua tools, script, dan data yang dibutuhkan telah tersedia di dalam folder SDBC, dengan struktur sebagai berikut:



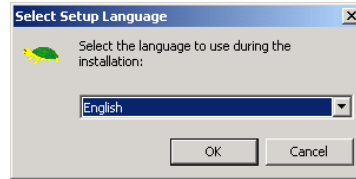
Di dalam folder tools, terdapat tools tambahan untuk pre-processing data: GrADS dan g2ctl. GrADS merupakan tool analisis dan visualisasi data meteorologi dan klimatologi yang umum digunakan, sementara g2ctl merupakan *tool* untuk menyiapkan data keluaran CFSv2 sehingga dapat diakses oleh GrADS. Sementara *tool* untuk proses statistical downscaling sendiri akan menggunakan Matlab®.

Sebagai contoh, dalam eksperimen ini digunakan keluaran CFSv2 tanggal 7 May 2016, dengan empat member *time-lag ensemble*:



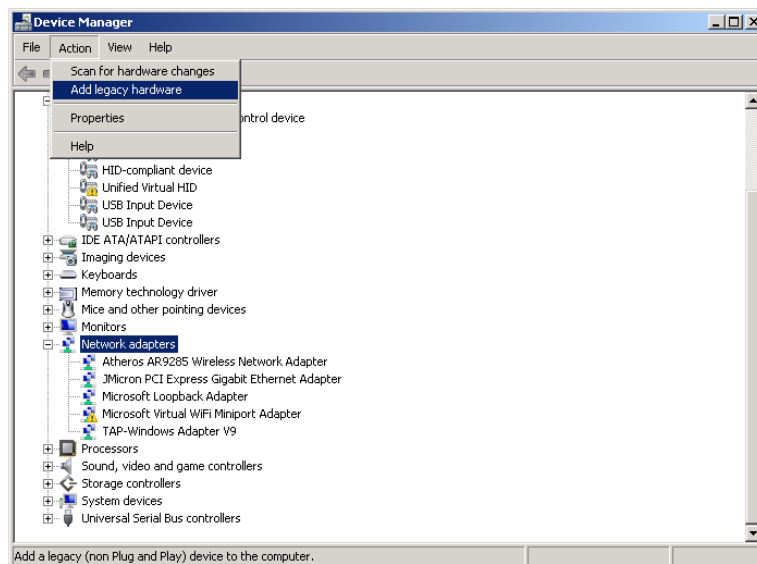
6.1.3 Instalasi dan Setting-up GrADS

Source program GrADS terletak pada folder `SDBC\Tools\OpenGrADS`. Untuk meng-install grads, cukup double-click file `grads-2.0.2.oga.2-win32_superpack.exe` lalu ikuti petunjuk instalasi hingga selesai:

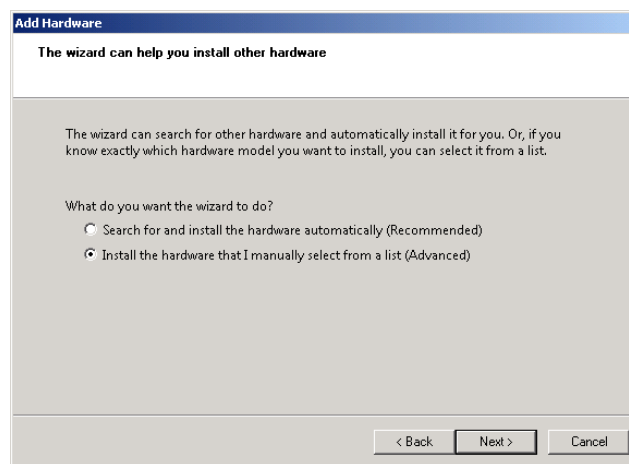


Setelah GrADS ter-install dengan sempurna, selanjutnya setup virtual adapter di Windows agar GrADS dapat dipanggil tanpa terkoneksi ke jaringan. Langkah-langkahnya adalah sebagai berikut:

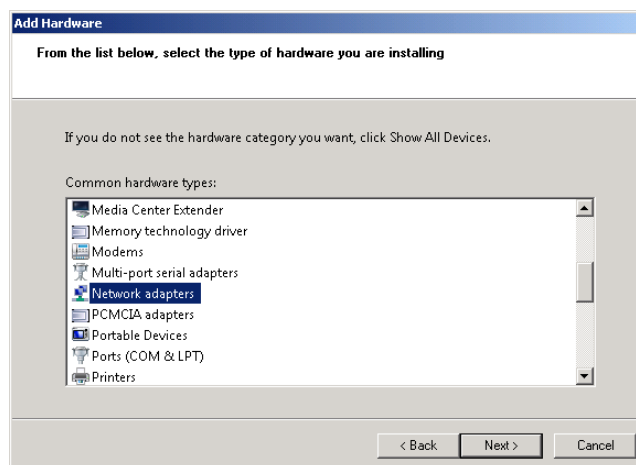
1. Buka Control Panel dan double click Device Manager
2. Click Network Adapter, Action-> Add legacy Hardware



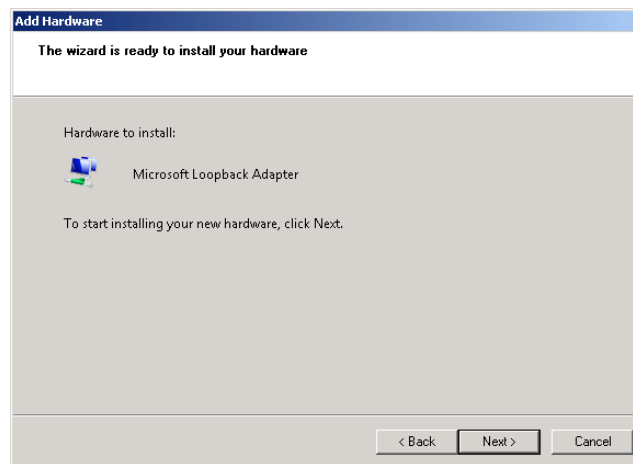
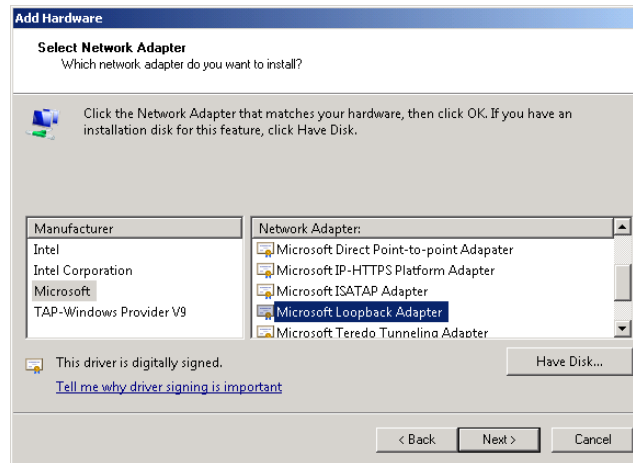
3. Next -> install manually -> Next



4. Pilih Network Adapter -> Next



5. Pilih Microsoft (di kiri) dan Loopback Adapter (di Kanan) -> Next -> Next



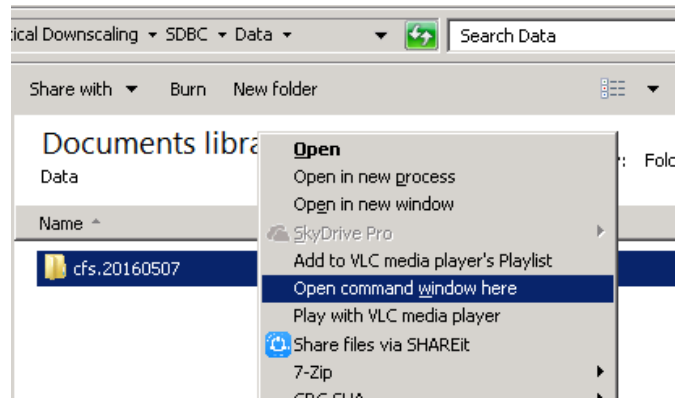
Selanjutnya, copy semua isi folder SDBC\Tools\g2ctl ke alamat:
C:\OpenGrADS\Contents\Cygwin\Versions\2.0.1.oga.1\i686

6.1.3 Pre-Processing Data Keluaran Cfsv2

Seperti yang dijelaskan sebelumnya, di eksperimen ini akan digunakan data prediksi operasional CFSv2 keluaran tanggal 7 May 2016 untuk target prediksi periode Juni-Juli-Agustus 2016. Secara default, keluaran CFSv2 disimpan dalam format GRIB versi 2, suatu format standar meteorologi. Agar lebih mudah diproses oleh Matlab, perlu diubah terlebih dahulu ke format binary biasa. Kita akan menggunakan GrADS dan g2ctl untuk itu.

Langkah-langkahnya adalah sebagai berikut:

1. Masuk ke folder Data
2. Shift + Right Click folder cfs.20070507, pilih Open command window here



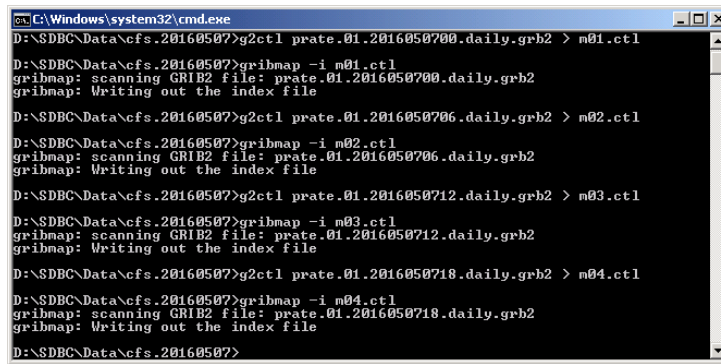
3. Di window Command Prompt Ketik perintah:

```
> g2ctl prate.01.2016050700.daily.grb2 > m01.ct1  
> gribmap -i m01.ct1
```

untuk member ke-1,

```
> g2ctl prate.01.2016050706.daily.grb2 > m02.ct1  
> gribmap -i m02.ct1
```

untuk member ke-2, dst hingga member ke-4



Harap diingat, bahwa data yang diunduh dari server CFSv2 adalah prediksi global, sehingga akan lebih baik jika datanya diekstrak terlebih dahulu untuk domain Indonesia saja. Kita bias lakukan itu dengan menggunakan GrADS. Langkah-langkahnya adalah sebagai berikut:

1. Kembali ke folder SDBC

Shift + Right Click folder Script, pilih Open command window here

2. Panggil GrADS dengan mengetikkan perintah berikut di command prompt:

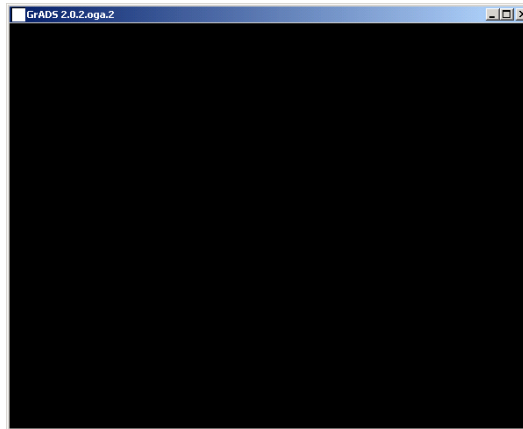
```
> grads
```

lalu tekan Enter dua kali untuk membuka GrADS dalam mode portrait. Jika GrADS telah terinstall dengan sempurna, kita akan mendapatkan dua window baru yang terbuka, GrADS command prompt dan display window.

```
c:\grads
D:\SDBC\Script>grads
Starting X server under C:\OPENGR\1\Contents\Resources\Xming
Starting grads under C:\OPENGR\1\Contents\Cygwin\Versions\2020GA*1.2\i686 ...

Grid Analysis and Display System (GrADS) Version 2.0.2.oga.2
Copyright (c) 1988-2011 by Brian Doty and the
Institute for Global Environment and Society (IGES)
GrADS comes with ABSOLUTELY NO WARRANTY
See file COPYRIGHT for more information

Config: v2.0.2.oga.2 little-endian readline printin grib2 netcdf hdf4-sds hdf5 o
pendap-grids:scn athena geotiff shapefile
Issue 'q config' command for more detailed configuration information
Loading User Defined Extensions table </cygdrive/c/OPENGR/1/Contents/Cygwin/Vers
ions/2020GA*1.2/i686/gex/udxt> ... ok.
Landscape mode? (<n' for portrait):
Gr Package Initialization: Size = 11 8.5
ga->
```



3. Jalankan script `S01_Ekstrak_Data.gs` dengan mengetikkan nama scriptnya ke command prompt GrADS. Jika berhasil dieksekusi, script akan mengekstrak data prediksi CFSv2 domain Indonesia untuk periode Jun-Jul-Agust 2016 ke dalam file `m01.dat` hingga `m04.dat`.

6.1.4 Eksperimen Statistical Downscaling

Pada bagian ini kita akan mulai eksperimen metode Statistical Bias Correction. Saat ini seharusnya dalam folder script telah tersedia file-file berikut (hasil dari tahap pre-processing):

- lon.dat dan lat.dat: merupakan informasi bujur dan lintang domain Indonesia
- m01.dat, m02.dat, m03.dat, dan m04.dat (member timelag ensemble CFSv2)

Untuk eksperimen ini, kita akan menggunakan script-script Matlab yang telah disediakan: `S02_Baca_Data.m`, `S03_Regrid`, dan `S04_Bias_Corr.m`. Script-script tersebut merepresentasikan tahapan eksperimen ini.

Eksperimen downscaling dibagi menjadi beberapa tahap:

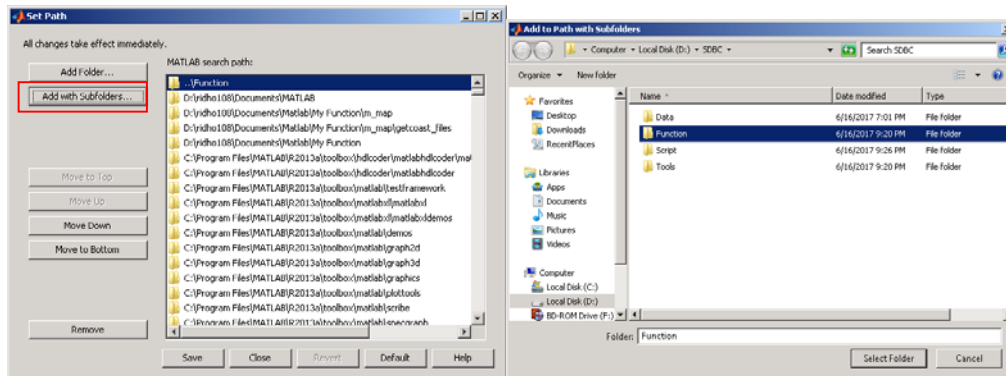
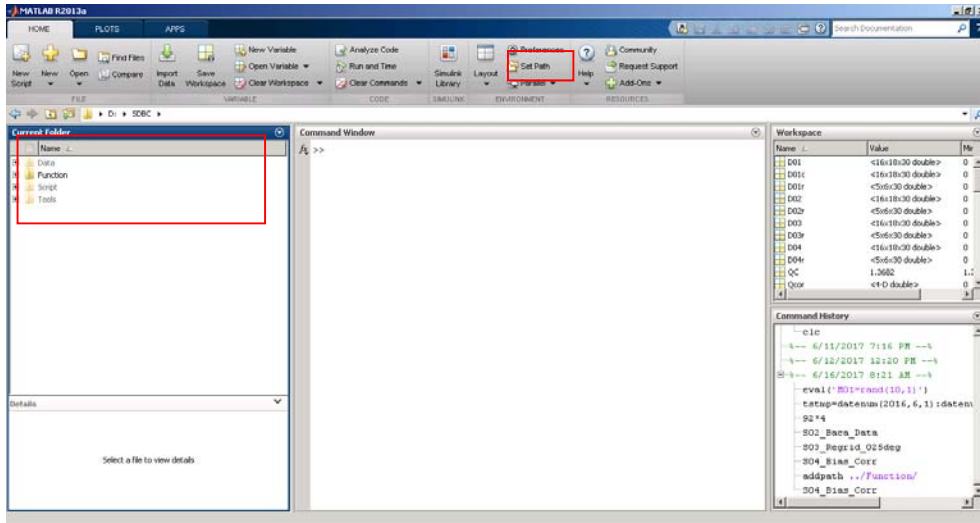
Tahap 1: Setting Path Matlab

Sebelumnya kita harus men-set up path environment Matlab. Ikuti langkah-langkah berikut:

1. Buka Matlab dan pindahkan Current Folder ke folder `SDBC/Script`
2. Tambahkan folder `Function` ke dalam path Matlab:

Diktat Pelatihan SD (Statistical Downscaling)

Home -> Set Path -> Add with Subfolders...



Tahap 2: Statistical Downscaling

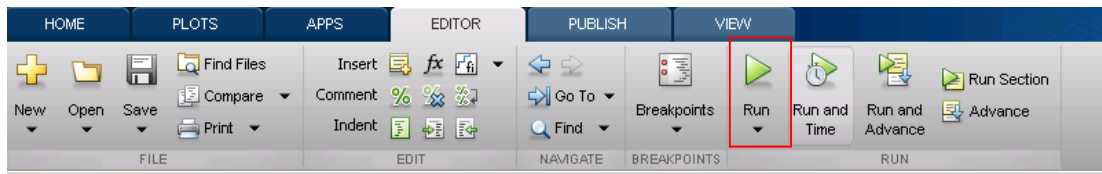
Tahapan statistical downscaling dilakukan dengan langkah-langkah berikut:

1. Double click script **S02_Baca_Data.m**. Script akan otomatis terbuka di window editor. Script ini bertugas membaca data keluaran CFSv2 (m01-m04), kemudian menghitung curah hujan akumulasi harian dan bulanan. Di script ini harap perhatikan **baris ke-3**.

Karena dalam eksperimen ini target prediksi adalah bulan juni-Agustus 2016, pastikan bahwa vektor tanggal yang tertulis di baris ketiga merepresentasikan periode tersebut.

```
S02_Baca_Data.m x
1      %% Load data dan hitung akumulasi harian serta bulanan
2      clear all;close all;clc;
3      tstmp=datetime(2016,6,1):datetime(2016,8,31); %vektor tanggal
4      %informasi lon dan lat
5      fid=fopen('lon.dat','rb');
```

2. Jalankan script tersebut dengan meng-click icon run di toolbar bagian atas (atau tekan tombol F5)



3. Setelah script S02_Baca_Data.m selesai running, akan muncul dua file berikut: data1dy.mat dan data1mo.mat
4. Langkah berikutnya adalah menjalankan script **S03_Regrid.m**. Script ini bertugas me-regrid ulang data CFSv2 (yang beresolusi ~100 km²) ke resolusi database observasi kita (TMPA, resolusi ~25 km²). Yang perlu diperhatikan dalam script ini adalah **baris ke-19**.

Di sini kita menggunakan fungsi `interp2` dengan opsi interpolasi `'nearest'`. Selain metode `nearest-neighbour`, Matlab juga menyediakan opsi interpolasi `'linear'`, `'spline'`, dan `'cubic'`.

```
16 -   for it=1:length(tstmp)
17 -       % interpolasi
18 -       d0=squeeze(dum(:,:,it));
19 -       zi=interp2(lonCFS,latCFS,d0,intlon,intlat,'nearest');
20 -       zi(zi<0)=0;
21 -       d0_int(:,:,it)=zi;
22 -       clear d0 zi
23 -   end
```

Jalankan script ini dengan tombol run atau F5. Keluaran script ini adalah file `dataaldy_regrid.mat`

- Langkah terakhir adalah menerapkan Bias Correction pada data keluaran CFSv2. Hal ini dilakukan oleh script **S04_Bias_Corr.m**. Beberapa hal yang harus diperhatikan pada script ini adalah:

Baris 3, 4, dan 5

Metode Bias Correction pada dasarnya adalah koreksi terhadap distribusi data. Oleh karena distribusi curah hujan berbeda tiap bulan (karena pengaruh musim), maka metode ini diterapkan pada masing-masing bulan target prediksi.

Baris 3, 4 dan 5 masing-masing merepresentasikan bulan yang akan dikoreksi: Juni, Juli, atau Agustus. Buka/tutup baris yang diinginkan dengan menghapus atau menambahkan simbol **%** pada awal baris.

```
1 - clear all;close all;clc;
2   % PER-BULAN TARGET PREDIKSI
3   % it=[1 30];itm=6;
4 - it=[31 60];itm=7;
5   % it=[61 92];itm=8;
```

Baris 11 dan 26

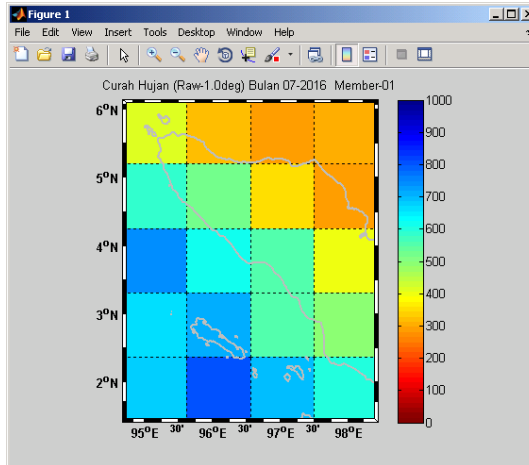
Dua baris ini masing-masing merepresentasikan subset domain Indonesia yang akan dikoreksi. Sebagai contoh, disini wilayah yang akan ditampilkan adalah wilayah Sumatera bagian utara. Edit kedua baris tersebut untuk merepresentasikan wilayah yang dipilih.

```
7   %% DATA Raw REGRID
8 - load dataaldy_regrid.mat
9   %Contoh: Wilayah Sumatera bagian utara
10 - ix=[1 16];iy=[33 50];
11 - lon=lonTRMM(ix(1):ix(2));
12 - lat=latTRMM(iy(1):iy(2));
13   %Member 1
14 - D01=d01_int(ix(1):ix(2),iy(1):iy(2),it(1):it(2));
15   %Member 2
16 - D02=d02_int(ix(1):ix(2),iy(1):iy(2),it(1):it(2));
17   %Member 3
18 - D03=d03_int(ix(1):ix(2),iy(1):iy(2),it(1):it(2));
19   %Member 4
20 - D04=d04_int(ix(1):ix(2),iy(1):iy(2),it(1):it(2));
21
22   %% DATA RAW ASLI
23 - load dataaldy.mat
24   %Contoh: Wilayah Sumatera bagian utara
25 - ix=[1 5];iy=[9 14];
26 - lonr=lonCFS(ix(1):ix(2));
```

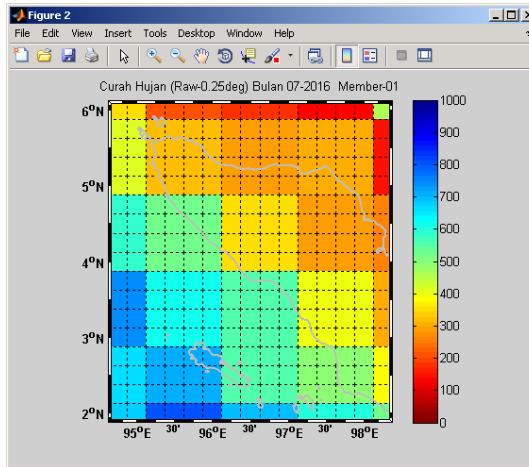
Baris 69 dst.

Baris ke 69 hingga baris terakhir adalah bagian untuk menggambar sampel hasil downscaling. Dalam contoh ini yang digambar adalah prediksi curah hujan bulan Juli 2016 member ke-1 CFSv2 Raw, CFSv2 Regrid dan CFSv2 Koreksi:

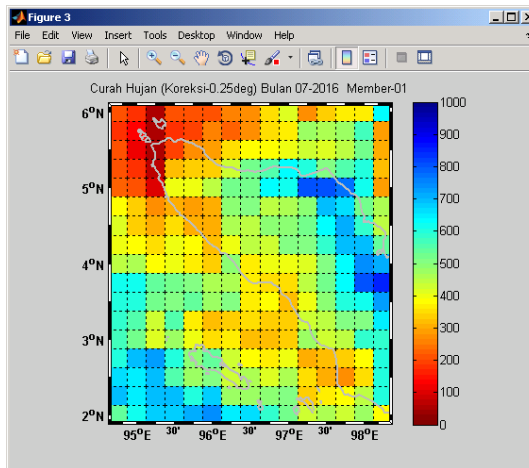
CFSv2 RAW (~100km²)



CFSv2 Regrid (~25km²) Nearest Neighbour



CFSv2 Bias Correction (~25km²)



6.1.5 Eksperimen Constructed Analogue

Pada bagian ini kita akan mulai eksperimen metode Constructed Analogue. Saat ini seharusnya dalam folder script telah tersedia direktori dan file-file berikut:

- Direktori **Data** yang berisi direktori a) Database Prediktan b) Database Prediktor c) Target Prediktan (Hasil CA) dan d) Target Pediktor
- Direktori **Function** berisi fungsi-fungsi yang dibutuhkan dalam eksperimen metode Constructed Analogue
- Direktori **ScriptCA** berisi script yang akan dioperasikan untuk menghitung dan mengolah data dalam eksperimen Constructed Analogue.

Untuk eksperimen ini kita menggunakan script-script Matlab yang telah disediakan: **S00_ContohCA.m**, **S00_PreProcessingTarget.m**, **S01_RunningCA.m**, **ANL01_HistoricalAnalysis.m**, **ANL02_ProbabilityForecast.m**, **ANL03_EkstrakData2TXT.m**. Script-script tersebut merepresentasikan tahapan eksperimen ini.

Eksperimen downscaling dibagi menjadi beberapa tahap, diantaranya:

Tahap 1: Statistical Downscaling Metode Constructed Analogue

Tahapan eksperimen constructed analogue ini dilakukan dengan langkah-langkah berikut:

1. Double click script **S00_ContohCA.m**. Script akan otomatis terbuka di window editor. Script ini bertugas membaca dan mengolah data prediktor dan prediktan yang digunakan untuk mengoperasikan eksperimen statistical downscaling constructed analogue.

```
1 - clear all;clc;close all;
2 - %contoh analog. untuk proses diagnosis (perbandingan pola) dan
3 - %prognosis (regresi) langsung digunakan data angin
4 - indir= '..\Data\';
5 - imdir='..\Images\';
6 - %folder fungsi2 yang dipakai
7 - addpath ..\Function\
8 - addpath ..\Function\m_map\
9
10 - tcon=30; %waktu sample (untuk flag=1)
11 - varn='strfunc'; %variabel prediktor (phi atau psi)
12 - mm=5; %bulan keluaran cfsr
13 - yy=2016; %target prediksi
14 - iw=1; %id window (1-5)
15 - iq=1; %id quartil (1-3)
16 - delta=15; %panjang pool database
17 - nbest=30; %banyaknya analog yang diambil
18 - smethod='cosim'; %metode similaritas yang dipakai dalam proses d
19 - rmethod='wmean'; %metode regresi yang dipakai dalam proses prognosi
20
```

Baris ke-11 merupakan bagian untuk mengatur script skema CA yang digunakan, pada **baris ke-18 dan ke-19** metode diagnosis dan prognosis yang digunakan serta pada **baris ke-16** mengatur panjang pool database yang digunakan. Pool database merepresentasikan jendela waktu yang dianggap mempengaruhi variabilitas prediktor target (default +/- 15 hari).


```

63 %=====ANALOG=====
64 %pool database di time window +/-delta hari di sekitar waktu target
65 idpool=fpool(t_time(it),d_time,delta);
66 pool_ptor=d_ptor(:, :, idpool);
67 pool_time=d_time(idpool);
68 %-----DIAGNOSIS-----
69 %pilih nbest analog terbaik dari pool database
70 warning('off','all');
71 [best_ptor,best_id,best_time,simv]=fdiag(t_ptor(:, :, it),pool_ptor,);
72
73 %-----PROGNOSIS-----
74 %buat regresinya
75 [koef_reg,r_ptor]=fprog(t_ptor(:, :, it),best_ptor,rmethod);
76 %-----CONSTRUCTED PREDICTAND-----
77 %30 best data base predictand
78 for i=1:length(best_time)
79     id(i)=find(d_time==best_time(i));
80 end
81 best_ptan=d_ptan(:, :, id);
82

```

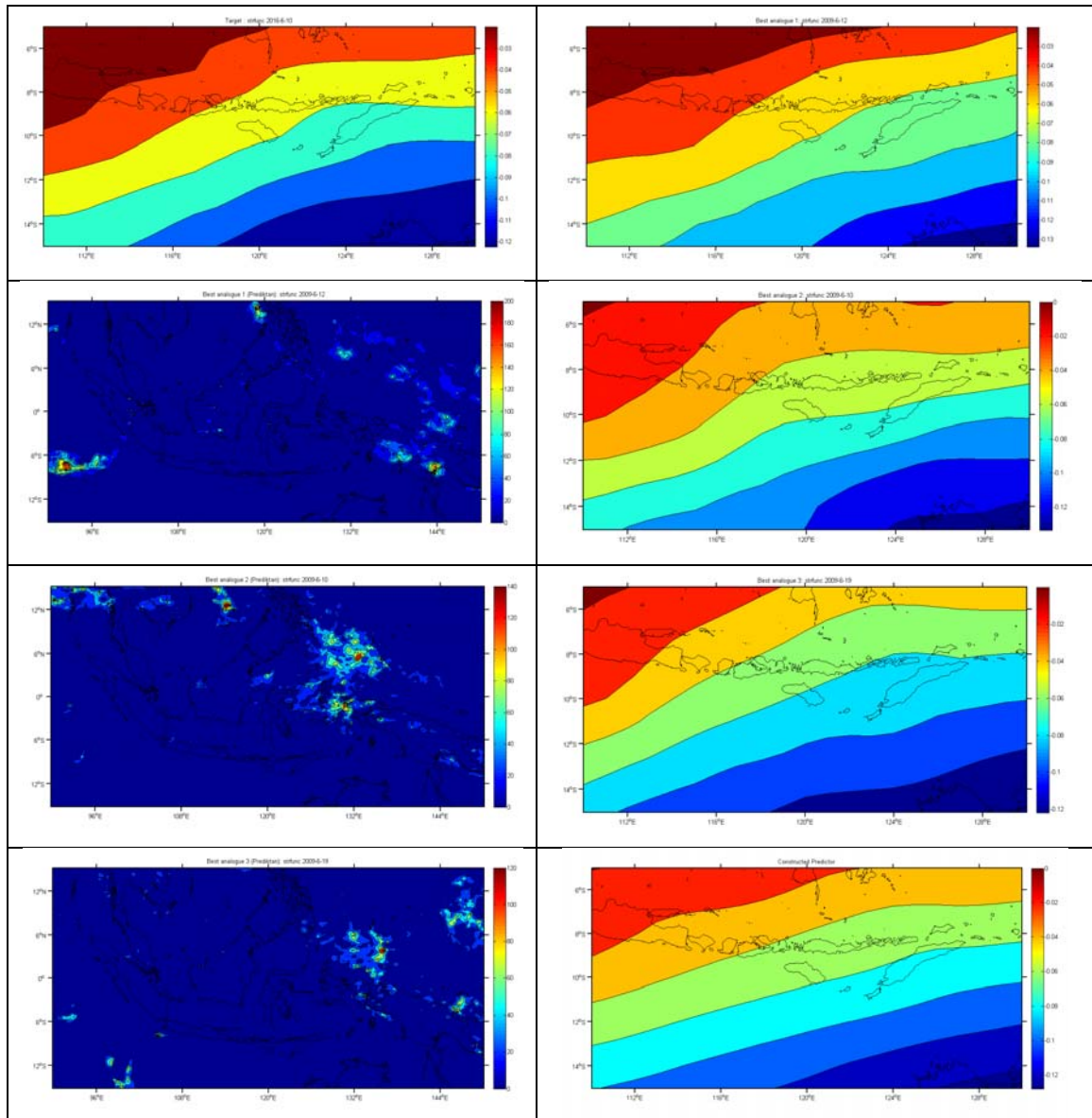
Baris ke-63 s.d baris ke-98 merupakan bentuk perhitungan analog dari eksperimen menggunakan metode constructed analogue. Kemudian untuk baris ke-106 s.d baris terakhir merupakan fungsi visualisasi Contoh CA.

```

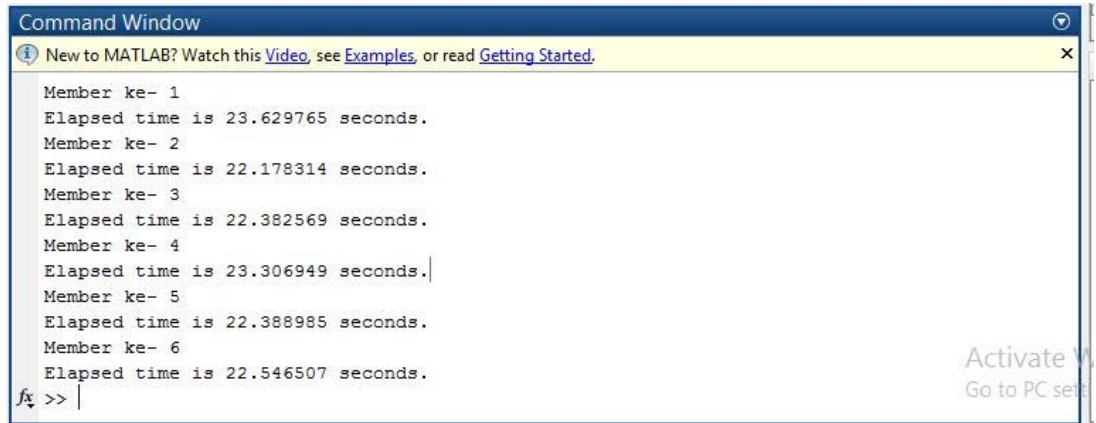
106 %bagian gambar2
107 switch flag
108 case 1 %hanya satu waktu, buat ngecek hasil pemilihan analog prediktornya
109     close all;
110     %Gambar analog nya
111     figure;
112     c1=min(min(t_ptor(:, :, tcon)));
113     c2=max(max(t_ptor(:, :, tcon)));
114     caxis([c1 c2]);
115     hc=fmap(t_ptor(:, :, tcon),lon,lat,'ina_coast',[c1 c2]);
116     ttl=datevec(t_time(tcon));
117     title(['Target : ' varn ' ' num2str(ttl(1)) '-' num2str(ttl(2)) '-' num2str(ttl(3))]);
118     %saveas(gcf,[mdir 'target.png'],'png');close;
119     dum=d_ptor(:, :, id);
120     for i=1:3*nbest
121         %ANALOG PREDIKTOR
122         figure;
123         hc=fmap(dum(:, :, i),lon,lat,'ina_coast',[c1 c2]);
124         ttl=datevec(best_time(i));
125         title(['Best analogue ' num2str(i) ' : ' varn ' ' num2str(ttl(1)) '-' num2str(ttl(2)) '-' num2str(ttl(3))]);
126         %saveas(gcf,[mdir 'best' num2str(i) '-prediktor.png'],'png');close;
127
128         %ANALOG PREDIKTAN
129         figure;
130         hc=fmap(best_ptan(:, :, i),lon_ptan,lat_ptan,'ina_coast',[0 50]);
131         ttl=datevec(best_time(i));
132         title(['Best analogue ' num2str(i) ' (Prediktan): ' | varn ' ' num2str(ttl(1)) '-' num2str(ttl(2)) '-' num2str(ttl(3))]);
133         %saveas(gcf,[mdir 'best' num2str(i) '-prediktan.png'],'png');close;
134     end
135
136     %gambar reconstructed prediktor
137     figure;
138     hc=fmap(r_ptor,lon,lat,'ina_coast',[c1 c2]);
139     title('Constructed Predictor');
140     %saveas(gcf,[mdir 'constructed_predictor.png'],'png');close;
141
142     %gambar constructed predictand
143     figure;hold all
144     ttl=datevec(t_time(tcon));
145     hc=fmap(r_ptan,lon_ptan,lat_ptan,'ina_coast',[0 50]);
146     title(['Constructed Analogue' num2str(ttl(1)) '-' num2str(ttl(2)) '-' num2str(ttl(3))]);
147     %saveas(gcf,[mdir 'constructed_predictand.png'],'png');close;
148 end

```

Hasil visualisasi untuk analog metode constructed analogue dari metode ini adalah sebagai berikut:

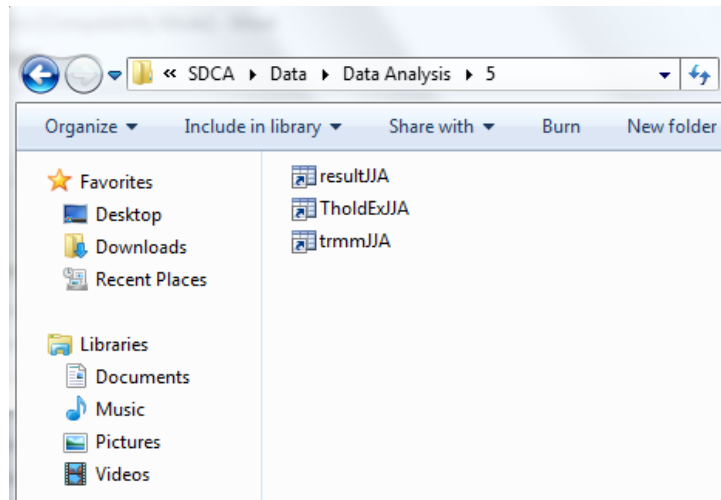


2. Kemudian operasikan script **S01_RunningCA.m**. Script ini mengoperasikan metode constructed analogue tanpa menggunakan proses filtering. Untuk proses diagnosis (perbandingan pola) dan prognosis (perata-rataan berbobot) langsung digunakan data psi dan phi. Jika sript S01_RunningCA.m dioperasikan akan menghasilkan proses untuk 6 member.

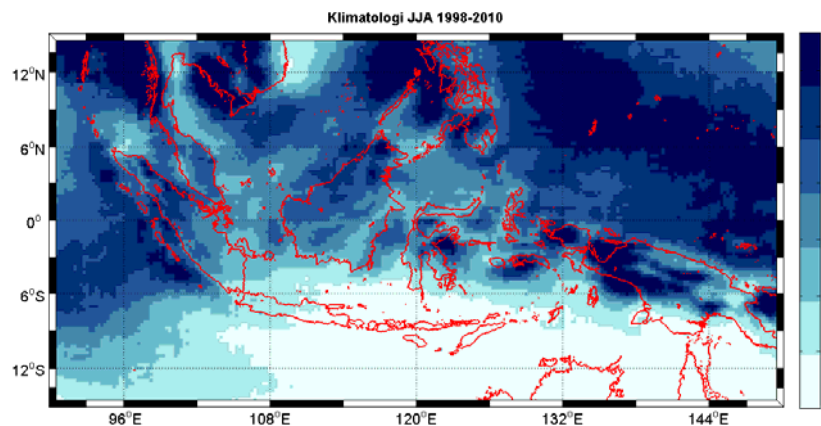
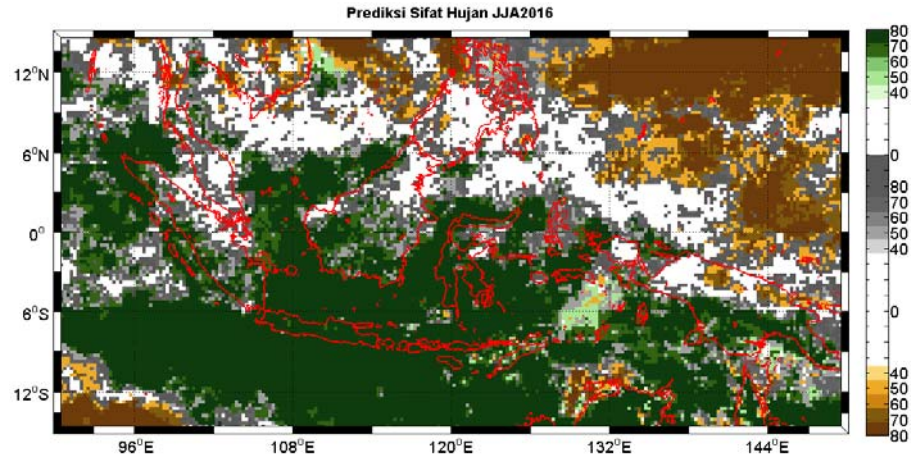


```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
Member ke- 1
Elapsed time is 23.629765 seconds.
Member ke- 2
Elapsed time is 22.178314 seconds.
Member ke- 3
Elapsed time is 22.382569 seconds.
Member ke- 4
Elapsed time is 23.306949 seconds.
Member ke- 5
Elapsed time is 22.388985 seconds.
Member ke- 6
Elapsed time is 22.546507 seconds.
fx >> |
```

3. Kemudian selanjutnya jalankan script **ANL01_HistoricalAnalysis.m**. Script ini mengoperasikan perhitungan untuk Rata-rata klimatologis hujan musiman (rata-rata untuk bulanan). Dengan output berupa data **resultJJA.mat**, **TholdExJJA.mat**, dan **trmmJJA.mat**.



4. Langkah selanjutnya adalah menganalisis hasil keluaran statistical downscaling constructed analogue dengan menggunakan metode analisis *probability forecast*. Metode ini dioperasikan dengan menjalankan script **ANL02_ProbabilityForecast.m**. Ouput yang dihasilkan dari mengoperasikan script ini adalah visualisasi peta sifat hujan dan peta distribusi klimatologis hujan.



5. Langkah terakhir yang dijalankan untuk menyelesaikan metode statistical downscaling ini adalah dengan menjalankan script **ANL03_EkstrakData2TXT.m**, yang bertujuan untuk mengekstrak data output dari keluaran hasil stastical downscaling ke dalam data ASCII. Output ekstrak data ini diantaranya: 1) **CA20165-sta-1.txt** 2) **CA20165-sta-2.txt** dan 3) **CA20175-sta-1.txt**. Display data seperti ilustrasikan oleh gambar di bawah ini:

CA20175-sta-1.txt									
1	Bujur:107.625								
2	Lintang:-6.875								
3	2017	6	1	3.4017	2.5484	2.719	2.2295	2.3528	4.8838
4	2017	6	2	2.3357	2.3965	3.0471	2.1902	3.3796	4.4917
5	2017	6	3	2.4678	2.5536	2.8611	2.7085	4.3007	2.5536
6	2017	6	4	3.5586	3.0589	2.9881	1.987	3.0515	4.3291
7	2017	6	5	2.666	2.375	3.0142	1.798	3.948	4.4322
8	2017	6	6	2.7228	3.4059	2.5053	1.8523	4.8875	3.7895
9	2017	6	7	3.0656	3.3146	2.8151	0.59107	4.3887	4.336
10	2017	6	8	3.5271	3.4956	3.7603	0.30265	5.0146	6.2414
11	2017	6	9	3.3062	2.6156	3.632	4.4572	3.8647	4.2564
12	2017	6	10	2.8754	2.825	3.0334	1.014	2.8185	2.7654
13	2017	6	11	1.9494	2.9906	3.8583	3.1398	4.825	4.2309
14	2017	6	12	1.9265	2.6789	3.8063	1.9631	3.6963	2.7127
15	2017	6	13	1.3091	1.7036	4.3015	1.8146	2.9883	6.3717