R-GUI

Mendesain Paket Analisis dan

Media Pembelajaran Statistika

File K Buka	irim Skrip	
Simpa Kelua	an Skrip r	
y<-2*x+mo z<-2*x+mo print(x) print(y) pairs(x,y,z) plot(x,y)	An exceeded and the second sec	
	QQ-Homermal Asil	CG-Homerrial Standar Standard Stan

I Made Tirta

Laboratorium Statistika-FMIDA Universitas Jember 2006

2006

PRAKATA

Puji syukur dipanjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmatNya jugalah buku ini dapat terselesaikan.

Saat ini referensi program R dalam bahasa Indonesia masih sangat langka, padahal program R sebagai *open source* menawarkan banyak kelebihan dibandingkan dengan program komersial yang harganya relatif mahal. Karena harga program komersial relatif mahal, di lapangan, bahkan termasuk di kalangan akademisi, program statistika yang beredar kebanyakan juga program yang tidak legal. Buku ini merupakan petunjuk praktis bagi pembaca (baik mahasiswa maupun dosen) yang ingin mengembangkan Program antar muka pengguna grafis (GUI *Graphical User Interface*) dengan menggunakan program R, khususnya untuk keperluan mendesain program analisis data atau media pembelajaran statistika. Pada bagian terakhir diuraikan cara mengemas program menjadi paket yang dapat didistribusikan sehingga dapat dimanfaatkan orang lain.

Besar harapan agar buku ini memberi manfaat bagi para pembaca yang ingin mengembangkan atau mendesain program analisis maupun program simulasi pembelajaran statistika. Saran dan kritik pembaca demi kelengkapan dan kesempurnaan buku ini ditunggu dan diterima dengan tangan terbuka.

Jember, Januari 2006 I Made Tirta

I Made Tirta

2006

DAFTAR ISI

PRAK	ATA		3
DAFT	AR ISI		5
DAFT	AR GAM	IBAR	8
1	TINJAL	JAN UMUM R	1
	Tujuan	Umum Error! Bookmark not defined	١.
	Tujuan	Khusus Error! Bookmark not defined	١.
	1.1 Fitu	ur Umum	1
	1.2 Fitu	ur Menu Versi WINDOWS	3
	1.3 Pak	ket RGUI-Rcommander (Rcmdr)	5
	1.3	3.1 RCommander untuk Versi Windows	5
	1.3	3.2 Rcommander versi LINUX	7
	1.4 Pak	ket untuk Mendesain R-GUI	8
	1.5 Edi	tor Skrip dengan Tinn-R 1	0
	1.6 Kei	untunganMenggunakan <i>Open Source</i> R1	4
	1.7 Bet	perapa Paket Terkait GUI1	6
	1.8 Bac	caan Lebih Lanjut1	7
	1.9 Soa	al-Soal Latihan 2	0
2	KOMPC	DNEN R-GUI2	1
	Tujuan	Umum Error! Bookmark not defined	I.
	Tujuan	Khusus Error! Bookmark not defined	Ι.
	2.1	Widget dengan Elemennya 2	1
	2.2	Menempatkan Elemen Widget 2-	4
	2.3	Bacaan Lebih Lanjut 2	6
	2.4	Soal-soal Latihan 2	6
3	DESAIN	N WIDGET SEDERHANA 2	7
	Tujuan	Umum Error! Bookmark not defined	Ι.
	Tujuan	Khusus Error! Bookmark not defined	I.
	3.1	Membuat Jendela sederhana 2	7

I Made Tirta

	3.2	Jendela dengan Tombol	32
	3.3	Membuat Jendela Pesan	33
	3.3	8.1 Mengganti Label	38
	3.3	8.2 Menambah Kotak Cek dan Tombol Radio	40
	3.4	Jendela dengan Kotak Daftar	44
	3.5	Jendela dengan Kotak Isian	46
	3.6	Jendela dengan Penggeser/ Slider	48
	3.7	Bacaan Lebih Lanjut	50
	3.8	Soal-Soal Latihan	50
4	DESAIN	VWIDGET KOMPLEKS	51
	Tujuan	Umum Error! Bookmark not define	ed.
	Tujuan	Khusus Error! Bookmark not define	ed.
	4.1	Jendela dengan Bingkai	51
	4.2	Membuat Menu Tunggal	53
	4.3	Membuat Menu Bertingkat	55
	4.4	Membuat Jendela Teks	58
	4.5	Dialog dengan Sistem Operasi	60
	4.6	Widget dengan tcltk2	62
	4.6	5.1 Membuat Tips pada widget	63
	4.6	5.2 Variasi Tampilan Jendela dengan tcltk2	65
	4.7	Bacaan Lebih Lanjut	68
	4.8	Soal-soal Latihan	68
5	APLIKA	SI R-GUI	70
	Tujuan	Umum	70
	Tujuan	Khusus	70
	5.1	Membuat Jendela Dialog	70
	5.2	Jendela Editor Skrip Sederhana	71
	5.3	Jendela Skrip dengan Menu	73
	5.4	Editor Skrip dengan Menu Edit	77
	5.5	Membuat Grafik Dinamik	79
	5.6	Grafik Dinamik dengan tkrplot	85
	5.7	Grafik Tiga Dimensi dengan RGL	88
	5.8	Desain Media Pembelajaran Statistika	90
	5.8	.1 Mendesain Sendiri Media Pembelajaran Statistika	90
	5.8	Memanfaatkan Paket TeachingDemos	94

	5.9	Dari Menu ke CLI	96
6	MENGE	EKSPLORASI DAN MEMODIFIKASI PAKET	99
	Tujuan	n Umum Error! Bookmark not de	fined.
	Tujuan	n Khusus Error! Bookmark not de	fined.
	6.1	Mengeksplorasi Paket	99
	6.2	Membuat Widget dengan Paket PBSmodelling	100
	6.3	Menerjemahkan dan Memodifikasi Menu Rcmdr	106
	1.9	9.1 Menerjemahkan menu dan dialog RCommander	106
	1.9	9.2 Memodifikasi dan Menambahkan Menu RCommande	er 107
	1.9	9.3 Menambahkan Menu Analisis	110
	6.4	Bacaan Lebih Lanjut	112
	6.5	Soal-soal Latihan	118
DAFT	ar pus	ТАКА	119
INDE	ks peni	ULIS	121
INDE	KS SUB	JEK	132

I Made Tirta

DAFTAR GAMBAR

Gambar	1.1 Tampilan Console R asli dalam Bahasa Inggris dengan Menu
Gambar	1.2 Tampilan Console R dalam Bahasa Italia dengan Menu yang
	Sangat Sederhana
Gambar	1.3 Pengaturan pilihan Bahasa pada Properties Shortcut R 6
Gambar	1.4 Tampilan RCommander dengan menu masih berbahasa Inggris
Gambar	1.5. Tampilan Menu R-Commander Versi Windows (sudah Berbahasa Indonesia)7
Gambar	1.6. Tampilan Menu R-Commander Versi Linux dengan menu Berbahasa Indonesia
Gambar	1.7. Tampilan Menu R-Commander Versi Linux (Mandrake). Tampilan dan kemampuan sama dengan versi Windows 8
Gambar	1.8 Grafik Interaktif untuk Ilustrasi Distribusi Bivariat (Normal). Posisi Grafik langsung berubah (berputar kiri-kanan atas bawah) jika slider digeser
Gambar	1.9 Grafik Interaktif untuk Mengilustrasikan Konsep Teorema Limit Pusat
Gambar	1.10 Contoh Tampilan Editor Sederhana yang <i>Built in</i> dengan R
Gambar	1.11.Tampilan Editor Skrip Tinn-R Versi 1.18 ketika tidak ada R yang aktif
Gambar	1.12. Tampilan Editor Skrip Tinn-R Versi 1.18 Ketika Ada R yang Aktif
Gambar	1.13.Tampilan Tool Komputer Tinn-R yang mirip Windows Explorer

2006

Gambar	1.14.Tampilan Tool R card yang berisi Refernsi Singkat Fungsi- fungsi R
Gambar	1.15.Tampilan Tool R Explorer Tinn-R yang memuat daftar
a 1	objek-objek R yang ada di memori komputer
Gambar	1.16 Contoh Tampilan Menu GRASP-R
Gambar	1.17. Tampian Menu RCommander yang telah Dilengkapi Menu TeachingDemos
Gambar	1 18 Contob Tampilan Penguii Berbagai Widget dari Paket
Gambai	PBSModelling
Gambar	1.19.Tampilan GUI PBSModelling untuk Mencoba Berbagai
	Ilustrasi yang Disediakan
Gambar	1.20.Contoh Tampilan GUI PBSModelling untuk Regresi Linier dengan Ilustrasi Grafik yang Dihasilkan
Gambar	2.1 Jendela Dialog dengan Beberapa Elemen seperti Kotak
	Daftar, Kotak Edit dan Berbagai Tombol
Gambar	2.2. Jendela Dialog dengan Frame, Slider dan Beberapa Tombol
Gambar	2 3 Posisi Penempatan Elemen Widget menurut Arah Mata angin
Gambai	Sebagai Pengganti Arah Kiri Kanan Atas Bawah 25
Gambar	3 1 Tampilan Jendela Dialog Sebelum Diberi Nama 28
Gambar	3.2 Tampilan Jendela Dialog Sesulah Diberi Nama 29
Gambar	3.3 Tampilan Jendela Sederhana dengan Teks atau Label
Gumbu	Penutunan jendela dapat dilakukan dengan mengklik tanda
	(×) nada nojok atas jendela 30
Gambar	3.4 Tampilan lendela dengan Berbagai Ukuran Label dan lenis
Gambai	Fon 22
Cambar	3 5 Tampilan Jendela dengan Label dan Tombol Deputun
Gambai	Jendela 33
Gamhar	3.6 Contoh Tampilan lendela Pesan dengan Pesan Informasi 34
Gambar	3.7 Contoh Tampilan Jendela Pesan dengan Pesan Beruna
Gambal	Peringatan
Gambar	3.8. Contoh Tampilan Jendela Pesan dengan Pesan Berupa
	Pertanyaan
	,

I Made Tirta

Gambar	3.9. Contoh Tampilan Jendela Pesan dengan Pesan Berupa Pemberitahuan Kesalahan
Gambar	3.10. Tampilan Jendela dengan Tombol OK dan Batal
Gambar	3.11. Tampilan Jendela dengan Pesan yang Dihasilkan dari
Gambar	3.12. Jendela dengan Tombol Penggantian Label
Gambar	3.13. Jendela dengan Tombol Penggantian Label dan Tampilan Sesudah Penggantian Label
Gambar	3.14. Tampilan Jendela dengan Kotak Cek dan Jendela Pesan ketika Kotak Cek tidak Di-klik
Gambar	3.15. Tampilan Jendela dengan Kotak Cek dan Jendela Pesan Ketika Kotak Cek Di-klik
Gambar	3.16. Tampilan Jendela dengan Tombol Radio dan Jendela Pesan dengan Pilihan Setuju
Gambar	3.17. Contoh Tampilan Kotak Daftar
Gambar	3.18. Tampilan Jendela dengan Kotak Isian dan Jendela Pesan ketika Tombol OK Di-klik
Gambar	3.19. Tampilan Silder Tegak dan Datar. Nilai variabel dapat diubah dengan menggeser slider keatas bawah atau ke kiri kanan 49
Gambar	4.1. Tampilan Jendela dengan Beberapa Bingkai dan Variasinya.
Gambar	4.2. Tampilan Menu Sederhana dengan Pilihan Tanpa Submenu
Gambar	4.3. Tampilan Jendela dengan Menu yang Tidak Bisa Dilepas (Menu I) dan Menu yang Bisa Dilepas (Menu II) 57
Gambar	4.4. Tampilan Jendela Edit dengan Teks yang Bisa Diedit. Jendela ini berguna untuk menulis skrip program
Gambar	 4.5. Tampilan Jendela dengan teks yang hanya bisa dibaca dan tidak bisa diedit
Gambar	4.6. Tampilan Jendela Dialog untuk Membaca File
Gambar	4.7. Tampilan Jendela Dialog untuk Menyimpan File 62
Gambar Gambar	4.8. Tampilan Komentar Setelah Menyimpan File

2006

Gambar 4.10. Contoh Tampilan dengan Beberapa Tips. Jika mouse
digerakkan pada wilayah 1, wilayah 2, tombol Tutup dan
OK akan muncul tips yang berbeda yang menjelaskan apa
yang akan terjadi jika mouse di-klik
Gambar 4.11. Contoh Tampilan Jendela Notebook dengan tcltk2 dengan
Dua Frame. Pengaktifan salah satu bingkai akan
menamplikan pilinan yang berbeda (tidak ada tombol atau
Combar 4.12 Contob Tampilan Jondola Panol Window dengan teltka
dengen Denel Berorietnesi Horizontel den Vertikal
Gambar 5.1 Tampilan Jendela Skrin (a) dan Grafik yang Dibasilkan dari
Pengiriman Perintah yang Ada nada Jendela Skrip (b)
Gambar 5.2. Tampilan Jendela Skrip Editor vang Dapat Dipakai Mengedit
dan Mengirim Perintah ke R75
Gambar 5.3 Tampilan Paket Editor R Sederhana. Jendela ini mampu
mencatat, mengedit dan mengirim skrip serta dilengkapi
menu memanggil bantuan R76
Gambar 5.4 Tampilan Jendela Bantuan R dengan Format html
Gambar 5.5. Contoh Tampilan Jendela Editor dengan Beberapa Menu
79 Pilinan
Gambar 5.6 Conton rampilan Grank Dinamik. rampilan memuat jenuela Widgot cobagai pangatur parameter dan Jandala Crafik
tempat tampilan grafik
Gambar 5.7. Tampilan Grafik Persepektif dengan tkrplot. Posisi gambar
dapat diubah dengan menggeser-geser slider
Gambar 5.8. Contoh Gambar 3D dengan rgl. Pada tampilan ini dua posisi
yang sedikit berbeda, (a) dan (b), diperoleh dengan
mengerak-gerakkan mouse sambil menekan tombol mouse
sebelah kiri
Gambar 5.9. Contoh Gambar Objek-objek 3D dengan rgl
Gambar 5.10. Tampilan Widget Pengontrol dan Tampilan Grafik Distribusi
I dan Z
Gambar 5.11. Hustrasi Tos Dang Logam. Dengan rgi dapat diliustrasikan
animasi putaran atau gerakan uang logam

I Made Tirta

Gambar	5.12. Ilustrasi Tos Dadu. Denga	n rgl dapat diilustrasikan
	animasi putaran atau gerakan da	du
Gambar	5.13. Tampilan Jendela Data untuk An	alisis Regresi 97
Gambar	6.1 Widget Sederhana dengan PBSMod	Jelling102
Gambar	6.2 Tampilan Grafik Dihasilkan dari Wi	dget Sederhana103
Gambar	6.2 Tampilan Widget Kompleks dengar	n Paket PBSModelling107
Gambar	6.3. Kontributor Terjemahan RCmdr ke	e Berbagai Bahasa108
Gambar	6.4. Tampilan RCommander Baku dala	m Bahasa Indonesia108
Gambar	6.5. Tampilan RCommander yang Tela	ah Dimodifikasi110
Gambar	6.6. Tampilan Dialog GEE. Dialog ini m	emiliki dua item tambahan
	diabandingkan dengan dialog	GLM, yaitu klaster dan
	struktur korelasi	

2006

1 TINJAUAN UMUM R

Pada Bab ini akan dibahas keunggulan dan keuntungan menggunakan R sebagai program *Open Source*. Setelah membaca informasi pada bab ini, pembaca diharapkan mengetahui makna program *Open Source;* keuntungan menggunakan program R; dan mencari situs yang membuat informasi tentang proram R

1.1 Fitur Umum

R adalah paket analisis data, yang merupakan paket *open source* yang termasuk keluarga bahasa S. R mempunyai struktur bahasa yang sama dengan SPlus yang dikembangkan secara komersial. Program R ini sudah dilengkapi banyak kemampuan internal untuk menganalisis data maupun menampilkan grafik. Bahkan, **kemampuan grafik merupakan keunggulan menonjol dari R maupun S-Plus dibanding paket lainnya**. Dewasa ini R populer dipergunakan baik dibidang akademik maupun industri. Sebagai *open soure* R bersifat *multiplatform* (jalan di **Windows** dan di **Linux**) dan didukung oleh banyak programmer maupun statistisi yang tersebar di seluruh dunia yang *source*nya dapat diperoleh di internet dan diijinkan untuk dimodifikasi sesuai keperluan (R Development Core Team-RDCT, 2007, lihat juga Tirta, 2005a).

Pada dasarnya R berinteraksi secara CLI (*Command Line Interface*) yaitu dengan membaca perintah yang dikirim melalui Jendela R-Console. Untuk membantu pengguna yang biasa bekerja dengan menu, GUI (*Graphical User Interface*), berbagai macam menu telah dikembangkan oleh berbagai kelompok. Salah satu menu, **R-Commander** diadopsi dan dimodifikasi oleh Tim Open Source Laboratorium Statistika FMIPA Universitas Jember untuk

I Made Tirta

menghasilkan menu berbahasa Indonesia bagi Pemula dan GUI **SciViews** bagi Programer/Pengguna statistika khusus/lanjut. Berbagai kebutuhan analisis statistika yang banyak dipakai, telah diimplementasikan ke dalam menu R-Commander. Sementara itu sangat banyak analisis statistika lanjut yang juga siap dimanfaatkan melalui CLI.

Beberapa analisis yang sudah bisa dilakukan melalui menu **R-Commander** diantaranya:

- 1. **Impor/Ekspor Data** yang berasal dari SPSS, Minitab maupun yang berupa ascii (txt).
- Eksplorasi dan Penyajian Data. Data univariate seperti Ringkasan data, Tabel Kontingensi, berbagai diagram seperti Histogram, Box-Plot, Normal Plot, QQ-Plot. Eksplorasi data multivariate seperti Tabel Korelasi, Diagram Pencar (2 variabel) yang dilengkapi dengan Histogram atau Box-plot, Matriks Diagram Pencar (untuk lebih dari 2 variabel) yang digabung dengan Box-plot, QQ-Plot atau Histogram. Kita dapat mengeksplorasi data secara intensif sebelum memilih analisis yang sesuai.
- 3. Analisis Data Parametrik. Uji-t: Uji mean dan Beda Mean, Uji-F: untuk uji anava (baik satu maupun multi arah), Uji beda Proporsi. Regresi biasa, Model Linier Tergeneralisasi (GLM) untuk regresi dengan data tidak harus berdistribusi normal dan bisa mengakomodasi data kuantitatif maupun kualitatif. Semua dapat dilengkapi dengan ilustrasi grafik yang memudahkan interpretasi hasil. Juga dilengkapi dengan diagnostik regresi atau model linier serta prosedur pemilihan model.
- 4. Analisis Non Parametrik seperti uji Wilcoxon dan Kruskal-Wallis.

Analisis lanjut yang bersifat khusus dapat dilakukan melalui CLI (*Command Line Interface*) langsung atau melalui Editor Tinn-R dan SciViews, seperti **Bootstrap**, **Monte Carlo**,

GeoStatistika/Spasial, Survival, GAM (Generalized Additive Model), GEE (Generalized Estimating Equation), SEM (Structural Equations Model), SOM (Self Organizing Map) Advanced Multivariate EDA (PCA, Discriminant, Cluster & Correspondence Analysis), Tree Regression, SPC/QCC (Statistics Process Control/Quality Control Chart), Time Series, Geo-R (untuk bidang Geografi/Geologi) dan masih banyak lagi, yang sebagian besar dilengkapi ilustrasi grafik. Disamping itu kemampuan program (TclTk, TkRplot) juga memungkinkan kita menulis program dan membuat paket mini (dengan GUI) untuk keperluan analisis tertentu, maupun untuk pengajaran Statistika (beberapa diilustrasikan dalam Manual). Grafik Interaktif dengan TclTk dan TkRPlot sangat berguna untuk Penanaman Konsep Statistika.

1.2 Fitur Menu Versi WINDOWS

Untuk versi Windows Konsul (Console)-R sudah memiliki kemampuan R-GUI walau sangat sederhana. Konsul ini sesungguhnya telah dilengkapi dengan fasilitas untuk menulis dan mengeksekusi skrip, namun masih sangat sederhana. Saat ini, untuk versi Windows Konsul dapat disajikan dalam berberapa bahasa diantaranya Italia, Perancis, Jepang. Versi Bahasa Indonesia sedang dalam proses pengerjaan dan diharapkan 2008 telah tersedia versi bahasa Indonesia. Contoh tampilan dalam Bahasa Inggris dan Bahasa Italia dapat diliha pada

Gambar 1.1 dan Gambar 1.2.

I Made Tirta



Gambar 1.1 Tampilan Console R asli dalam Bahasa Inggris dengan Menu yang Sangat Sederhana



Gambar 1.2 Tampilan Console R dalam Bahasa Italia dengan Menu yang Sangat Sederhana

2006

1.3 Paket RGUI-Rcommander (Rcmdr)

Secara umum kemampuan R dikemas dalam bentuk kumpulan fungsi-fungsi yang disebut paket (*package*). Paket-paket yang telah diinstal berkumpul menjadi suatu pustaka (*library*) dari R, karena itu paket yang telah diinstal biasa disebut pustaka (*library*). Beberapa paket telah berbasis R-GUI, diantaranya yang populer adalah RCommander (Rcmdr) yang utamanya diperuntukkan bagi mereka yang baru belajar R dan telah biasa bekerja dengan menu.

1.3.1 RCommander untuk Versi Windows

Sebagaimana paket *open source* pada umumnya, menu Rcmdr juga dapat diterjemahkan ke Bahasa Indonesia. Modifikasi ke dalam Bahasa Indonesia telahdilakukan penulis sejak versi R 1.91 tahun 2004. Saat ini untuk versi 1.3 telah tersedia RCommander versi Bahasa Indonesia yang dapat diinstal scara otomatis dan dieksekusi dengan memberi pilihan Language=id pada short cutnya (Lihat Gambar 1.3). Jika tersedia Konsuldalam Bahasa Indonesia, pilhan ini sekaligus menentukan bahasa pada Konsul R maupun pada RCommander.

I Made Tirta

R 2.5.0 Target tope: Application Target location: Iarget m Flex\R\R.0.5.0\hin\Rgui exe"LAN3HA3E= Startin: "C:\Program Files\R\R-2.5.0" Shortout_key: None Burr	R 2.5.0 Target type: Application Target location: bin Iarget m Flex\R\R\2.5.0* Start in: "C:\Phogram Files\H\R_2.5.0" Shortcut gey: None Burr Normal window	R 2.50 Target type: Application Target location: bin Larget mr Flex/NR/\$ 5 0%bin/Rg.ai exe*'LAN3HA3E#ail Start in: "C:/Program Files/HVR/2.5.C" Shortout_sey: None Burr Normal window Cgmment.	General Shori	Cut Compatibility
R 2.50 Target type: Apploation Target location: bin Larget m Files/RNR.0.5000000000000000000000000000000000	R 2.50 Target type: Apploation Target location: bin Larget m Flex\R\R\$ 50\bin\Rguitexe"LANBILABE# Start int: "C:\Phogram Files\H\R\$2.5.C" Shortout Ley: None Bur Normal window	R 2.50 Target type: Application Target location: bin Larget m FleckNKR\$500binkRg.itexe*LANSHASE=it Start int: "C:VProgram FilesVRVR2.5.0" Shortcut key: None Burr Normal window Cgimment. Implication		and the support
Image type: Apploation Target location: bin Iarget m Flex\R\R∲ 5.0\bin\Rgui exe"LAN3HA3E= Start in: "C\Ptogram Files\FI\Flex\S.U" Shortcut key: None Burr Normal window	Image Transmission Application Target location: bin Target mr Flex\R\R⊉ 5.0\bin\Rguitexe*'LANBILABE= Start in: "C:\Ptogram Files\R\R2.5.0" Shortcutkey: None Burr Normal window	Image: type: Application Target location: bin Iarget: Image: Flex\B\B\$ 5 0%bin\Bgui exe*LANSUASE=in Start in: "C:\Phogram Files\H\R-2.5.C" Shortcut key: None Burr Normal window Cgimment. Image: Flexible window	R F	3 2.5.0
Target type: Apploation Target location: bin Iarget mr Flex\R\R∳50\bin\Rgui exe''LAN3HA3F= Start m: "C\Ptogram Files\H\H2.5.C" Shortcut key: None Burr Normal window	Target type: Application Target location: bin Iarget m Flex\R\R\\$ 50\bin\Rguitexe"LANBIA3E= Start in: "C:\Ptogram Files\R\R.2.5.C" Shortcutkey: None Burn Normal window	Target type: Apploation Target location: bin Target m Flex\R\R∲ 5 0\thin\Rgui exe*TAN3HA3E=in Start in: "C:\Phogram Files\H\R-2.5.C" Shortcut key: None Burr Normal window M Cgimment I		
Target location: bin Iarget mr Flex/RNR.\$5.0%bin/Rgui exe"LAN3HA3F= Start mr. "CVPtogram Files/HVH-2.5.0" Shortcut Key: None Burr Normal window	Target location: bin Iarget m Flex\R\R∲ 5.0\bin\Rguitexe*LAN3HA3Eei Start in: "C:\Ptogram Files\R\R2.5.0" Shortcutkey: None Burn Normal window	Target location: bin Larget m Flex\R\R∲50\bin\Rgui axe"LAN3HA3E=in Start m: "C:\Phogram Files\H\R-2.5.C" Shortcut key: None Burr Normal window b Cgmment I	Target type:	Application
Jarget m Flex\R\R∲ 5 0\bin\Rguitaxe"LAN3HA3E= Start int "C\Ptogram Files\R\R42.5.C" Shortcutigey: None Burr Normal window	Iarget m Flex\R\R∲ 5 0\bin\Rgui axa"LAN3HA3E=i Start in: "C\Program Files\R\R2.5.0" Shortcut Ley: None Burr Normal window	Larget m Flex\R\R\$ 50\thin\Rgui axa"LAN3HA3E=iit Start in: "C:\Ptogram Files\H\H-2.5.C" Shortcut key: None Burr Normal window Cgmment Image: Compare the second secon	Target locatio	n: bin
Shart out §ey: None Burr Normal window	Emilie "C:\Phogram Files\R\R-2.5.C" Shortcut Key: None Burr Normal window	Shart in: "C:\Ptogram Files\H\\R-2.5.C" Shart ut key: None Burr Normal window Cgimment.	Target	m Flex/B/B/ 5 0/bin/Boui ave" ANGUAGE
Start in: "C:\Program Files\PRP-2.5.C" Shortcut Key: None Burr Normal window	Start in: "C:\Phogram Files\RVR-2.5.0" Shortcutkey: None Burr Normal window	Start in: "C:Whogram FilesWHVF-2.5.0" Shortcut Key: None Burr Normal window Cgimment Image: Compared to the start of the star	Tailler	in the other particular and the contract of
Shartout key: Nore Burr Normal window	Shortcut key: Nore Burr Normal window	Shartcut Key: Nore Burr Normal window Cgmment	Start in:	"C\Program Files\B\B-2.5.0"
Shortcut key: None Burr Normal window	Shortcut key: Nore Burr Normal window	Shortcut <u>key:</u> Nore Burr Normal window Cgmment	<u>_</u>	
Bur Normal window	Burr Normal window	Burr Normal window S Cgmment	Shortcut <u>k</u> ey:	Nore
		Comment	Burr	Normal window
Connect	Concept	Common.	Connent	
			eginnen.	
	Eind Target Change Icon Advarced		Eine	d Target Change Icon Advarced

Gambar 1.3 Pengaturan pilihan Bahasa pada Properties Shortcut R

File Edit Data	Statistics Graphs Mod	els Distributions Tools Help	
Script Window	Summaries Contingency tables Means Proportions Variances Nonparametric tests Dimensional analysis	ta set View data set Nodel: <no active="" model=""></no>	
	Fitmodels		



2	n	n	6
2	υ	υ	υ

File Edit Data	Statistika Grafik Model Distrik	usi	Bantuan
Lab-Stat, FMIPA-L	Ringkasan Tabel kontingensi	*	at cata set Nodel: <mark>(Belum Aca Model)</mark>
data (Orange,	Proporsi	+	
attach (Orange	Varians	•	
scatterplot (c	Uji nonparametrik	•	line='m, smooth=FALSE, labe
abline(lm(cir	Regresi/ Model linier		<pre>black",ltv=1))</pre>
	Anava/ Rata-rata	•	Uji t Sampel Tunggal (mean)
	Multivariat/ Analisis Dimensional	•	Uji t Sampel bebas (beda mean)
	Transformasi	*	Ujit Berpasangan
-			ANAVA satu arah
			ANAVA-multi arah

Gambar 1.5. Tampilan Menu R-Commander Versi Windows (sudah Berbahasa Indonesia).

1.3.2 Rcommander versi LINUX

R adalah proram komputer yang multiflatform dan sebagian besar fitur grafis yang dimiliki versi windows juga dimiliki oleh versi Linux. Bahkan beberapa kemampuan R seperti RWeb saat ini lebih mudah difungsikan pada pada versi Linux atau Unix. Menu bahasa Indonesia pada RCommander juga dapat difungsikan pada versi Linux.

1.00	Edit Dela Staleiks Graik	Model Distribusi Eau	ituen - Stelistika Klu	reus/Den u			
R.	Lac-Gest, I Million UNLL. Data zet:	(Tuk 4da dala akli))	Lot data set	Linat data set	Mode:	(Tuk Ada Model Akli*)	4 ^{Submil}
							-

Gambar 1.6. Tampilan Menu R-Commander Versi Linux dengan menu Berbahasa Indonesia.

I Made Tirta





1.4 Paket untuk Mendesain R-GUI

Selain dilengkapi dengan berbagai paket untuk analisis data dan visualisasi Grafik, R juga memiliki beberapa paket yang dapat dipergunakan untuk mendesain GUI. Dalam buku ini pembahasan difokuskan pada paket yang bermanfaat untuk mengembangkan GUI. Paket-paket tersebut diantaranya adalah tcltk, tcltk2, tkrplot. Dalam kenyataan menu Rcommander juga memanfaatkan kemampuan paket tcltk. Selain untuk keperluan membuat menu program analisis, paket-paket ini (tcltk, tcltk2, tkrplot) juga dapat dimanfaatkan ntuk mendesain media pembelajaran statistika. Beberapa contoh media ini dapat dilihat pada dan Gambar 1.9.

2006



Gambar 1.8 Grafik Interaktif untuk Ilustrasi Distribusi Bivariat (Normal). Posisi Grafik langsung berubah (berputar kiri-kanan atas bawah) jika slider digeser.

I Made Tirta



Gambar 1.9 Grafik Interaktif untuk Mengilustrasikan Konsep Teorema Limit Pusat.

1.5 Editor Skrip dengan Tinn-R

Sebagian besar interaksi dengan R dilakukan melalui CLI (*Command Line Interface*), walau demikian R sendiri dilengkapi dengan paket yang dapat digunakan untuk mendesain GUI. Untuk mengoptimalkan interksi melalui CLI diperlukan editor skrip yang baik. R sendiri sebenarnya sudah dilengkapi dengan editor skrip yang sederhana. Editor ini telah dilengkapi dengan kemampuan *copy, cut, paste,* dan menjalankan sebagian atau seluruh skrip (lihat Gambar 1.10).

Untuk editor yang lebih lengkap, ada editor *open source* yang khusus dirancang untuk R, yaitu Tinn R (lihat Gambar 1.12. Tampilan Editor Skrip Tinn-R Versi 1.18 Ketika Ada R yang Aktif. . Tinn-R diprakarsai oleh

2006



Faria & Grosjean (2005) dengan kontribusi banyak orang. Ada beberapa keuntungan menulis skrip R dengan Tinn-R diantaranya:

Gambar 1.10 Contoh Tampilan Editor Sederhana yang Built in dengan R

- 1. memiliki kemampuan membedakan struktur skrip (*syntax highlighted*); adanya perbedaan warna untuk tiap kata kunci memudahkan kita memeriksa skrip progranm yang ditulis;
- 2. memilikikemampuan memeriksa kelengkapn pasangan kurung;
- 3. telah tersedia panel komunikasi dengan R (seperti mengirim skrip, sebagian atau seluruhnya; membersihkan jendela console)
- telah tersedia kartu referensi untuk beberapa fungsi mendasar; dengan kartu ini kita dapat dengan cepat melihat struktur fungsifungsi yag kita perlukan;
- 5. mampu memberikan petunjuk (*clue*) lengkap ketika suatu fungsi yangtelah didefinisikan dipanggil;



Gambar 1.11. Tampilan Editor Skrip Tinn-R Versi 1.18 ketika tidak ada R yang aktif.



Gambar 1.12. Tampilan Editor Skrip Tinn-R Versi 1.18 Ketika Ada R yang Aktif.

2006

	R explorer	R card	Tags	Project	Computer
-					10 50
-				.*)	\$ All (*
^				p	🕑 Deskto
			ents	Docume	🕀 🔛 M
			er Places	v Network	
~			1	ecycle Bir	
HIBA	in 🖯 TO	rVideo Wir	🗾 Inte	Explorer	🥭 Internet
Edt	art 🛃 Wi	o StartSma	. 🥘 Nei	Reader	Acrobat
endix-s	ss 🔀 apj	ac Wireles	C Nel	Acrobat	Adobe A
N	P DF	5.0	RR R 2	asy-Pri	🌏 Canon B
cises-	ion 🔁 Ex	/iews R Co	RSci	iotoPrint	🙆 Easy-Ph
•					<

Gambar 1.13. Tampilan Tool Komputer Tinn-R yang mirip Windows Explorer.

	41/	7 .			
Basic and h	elp				
Data (load, i Data (select Dates and ti Distributions	iead, writ ion and i mes	te and sa manipula	ave) ition)		
1<			•	>	
array(x, di	m=)				
c()					-
cbind()					
data.fram	e()				~
Arrau with d	ata x: sn	ecify dim	ensions lik	e dim=c(3.4.2)	. 2

Gambar 1.14. Tampilan Tool R card yang berisi Refernsi Singkat Fungsi-fungsi R.

I Made Tirta



Gambar 1.15. Tampilan Tool R Explorer Tinn-R yang memuat daftar objek-objek R yang ada di memori komputer.

1.6 KeuntunganMenggunakan Open Source R

Ada beberapa keuntungan apabila kita memanfaatkan program *open source*, diantaranya adalah seperti berikut ini.

1. **Legal dan murah**. Program dapat dikopi dan didistribusi secara bebas. Biaya penggantian terbatas pada jasa (mengkompilasi, memodifikasi, ongkos cetak dan kirim) dan manual dalam Bahasa Lokal (Bahasa Indonesia). Ini akan mengurangi ketergantungan pada Program Bajakan, sekaligus menghilangkan citra Bangsa Indonesia sebagai salah satu pembajak terbesar.

2006

- 2. Terbuka untuk dikembangkan dan dimodifikasi (cocok bagi mereka yang tertarik sebagai pengembang dan tidak sekedar pengguna). Bahkan dengan *library* TclTk, terbuka peluang untuk membuat paket-paket kecil untuk kebutuhan khusus baik untuk keperluan analisis data maupun pengajaran (penanaman) konsepkonsep Statistika (seperti Peluang dan Frekuensi relatif Teorema Limit Pusat, Interval Keyakinan dan Coverage Probability, Regresi & Outlier dan sebagainya).
- 3. **Multiplatform** dapat dijalankan baik di Windows maupun di Linux.
- 4. **Kemampuan Statistika dan Grafik dengan spektrum luas** dan hampir 100% *compatible* dengan Paket komersial S-Plus (Semua Pustaka/*Library* untuk S-Plus dapat diaplikasikan pada R. Sebagian besar kemampuan analisis sudah dapat diakses melalui menu berbahasa Indonesia yang sederhana dan mudah diikuti.
- 5. Didukung banyak statistisi kelas dunia yang tersebar di berbagai universitas di seluruh dunia. Buku teks tentang analisis statistika lanjut yang secara khusus menggunakan R atau S-Plus, diantaranya: *MASS (Modern Applied Statistics with S-Plus)* oleh Venables & Ripley; *GAM (Generalized Additive Models)* oleh Hastie & Tibshirani. *Statistical Model in S* oleh Chamber & Hastie dan *A Handbook of Statistical Analysis Using S-PLUS* oleh Everitt. Tersedia juga berbagai dokumentasi elektronik dalam format file pdf. Telah pula dirintis beberapa manual berbahasa Indonesia diantaranya *Buku Panduan Program Statistika R* (ISBN: 979-8176-37-5, dan dilengkapi CD R Versi 2.01 & 2.1.1); juga dalam proses penulisan, *Analisis Statistika dan Grafik dengan R*, yang juga tersedia dalam bentuk slide elektronik.
- 6. **Dapat menganalisis data melalui web.** Ada beberapa paket R yang dapa memfungsikan R sebagai web server. Paket-paket ini diantaranya adalah Rpad, RWeb, RCgi dan masih ada lagi yang lainnya. Sebagian besar paket ini hanya berfungsi pada versi Linux. Dengan paket sejenis ini, kita dapat menganalisis data menggunakan R sekalipun pada komputer tempat kita bekerja tidak ada program R. Melalui web kita dapat memangggil R dan mengirm data atau perintah lainnya untuk dianalisis R yang ada di server lain.

I Made Tirta

1.7 Beberapa Paket Terkait GUI

`Beberapa paket R (khususnya versi 2.5) telah mengembangkan dan memanfaatkan GUI. Berikut adaah beberapa paket R yang dapat dieksplorasi untuk lebih memahami pengembangan GUI untuk R. Gambar Gambar 1.16 sampai Gambar 1.20 mengilustrasikan tampilan GUI dari beberapa paket tersebut.

Tabel 1. Paket-paket R yang Memanfaatkan atau Mengembangkan GUI

No	Nama file	Nama Paket	Deskripsi
1	tcltk	Tcl/Tk	Paket khusus untuk
			mengembangkan RGUI
2	Tcltk2	Tcl/Tk2	Paket khusus untuk
			mengembangakan RGUI,
			pengembangan dari Tcl/Tk
3	tkrplot	TkRPlot	Paket untuk melengkapi
			ilustrasi grafik pada RGUI
4	Rcmdr	Rcommander	Paket Analiis Statistika yang
			dikembangkan melalui Tcl/Tk
5	grasper	GRASP-R	Paket khusus Analisis Regresi
_			Umum dan Analisis Spasial
6	PBSModelling	PBSModelling	Paket analisis data kelautan
			(Pasifik), tetapi banyak memuat
			fungsi untuk pengembangan
_		T 1. 5	RGUI
/	TeachingDemos	TeachingDemos	Animasi menggunakan GUI
			terkalt pembelajaran teori
			peluang dan statistika. Menu
			dapat digabung (<i>embedded</i>)

2006

1.8 Bacaan Lebih Lanjut

Referensi tentang R, terutama yang berbahasa Indonesia memang belum banyak. Bagi pembaca yang baru mengenal R disarankan membaca informasi yang lebih umum diantaranya Tirta (2005a). Referensi berbahasa Inggris cukup banyak tersedia di Internet. Sebagian besar referensi ini dapat dilacak pada situs <u>http://www.r-project.org</u>. Khusus untuk pengembangan R-GUI dapat dilihat referensi Grosjean (2006). Informasi lebih jauh tentang R dapat juga dilihat pada file FAQ (*Frequently Ask Questions*) dari *manual online* RDCT (2006).

GRASP for R	
ile Edit Analysis Plot Export Help	
GRASP for R	
Generalized Regression Analysis and Spatial Prediction	for R
/elcome to GRASP for R	
orm Mahad week to face	

Gambar 1.16 Contoh Tampilan Menu GRASP-R



Gambar 1.17. Tampian Menu RCommander yang telah Dilengkapi Menu TeachingDemos

I Made Tirta

		Ch	oose a	Wic	iget C CLEAR	
С	button	С	label	c	slider	
с	check	С	matrix	С	text	
C	data	С	menu	С	vector 1	
C	entry	C	null	C	vector 2	
C	grid	С	object	C	window (horizontal)	
C	history	•	radio	C	window (vertical)	
ndov d 1 4 radio	v title="\w 4 pady=1(name=ju	/idget 0 nk val	= radio'' ue=0 text=	=''Non	9"	
ndov d 1 4 radio radio radio radio	v title=''W 4 pady=10 name=iu name=iu name=iu name=iu	/idget 0 nk val nk val nk val	= radio'' ue=0 text= ue=1 text= ue=2 text= ue=3 text=	="Noni ="Optic ="Optic	9" on A" on B" on C"	
ndov id 1 4 radio radio radio	v title='\w 4 pady=1(name=ju name=ju name=ju name=ju	/idget 0 nk val nk val nk val nk val	= radio" ue=0 text= ue=1 text= ue=2 text= ue=3 text= = radio	=''Non =''Optic =''Optic	s" on A" on B" on C"	
ndov d 1 4 radio radio radio	v title=''w 4 pady=11 name=ju name=ju name=ju 7k Wi 1 C No	/idget 0 nk val nk val nk val nk val dget one (= radio'' ue=0 text= ue=1 text= ue=2 text= ue=3 text= = radio C Option	=''Noni =''Optic =''Optic	9" on A" on B" on C" © Dption B © Option C	

Gambar 1.18. Contoh Tampilan Penguji Berbagai Widget dari Paket PBSModelling

2006

choose an E	xample 👘	LLEAH
Random Variables	Statistical Analyses	Other Applications
C RanVars	@ LrReg	C LissFig
C BanProp	C MarkFiec	C FishRes
C SireNom	C CCA	C FishTows
C CacVor	C yonB	C YFB
	С бым	C Sudeku
		C TestFuna
rid I 6 sticku=W label taxt-View: Ioni button text-Elose fu button text-Model fu button text-Data fu button text-Modew button text-Window rid 3 1 lopttie-"Cho	H-Ecld nc=sperFile action= unc-openFile action or-openFile action- trunc-openFile action func-openFile action oce Example" topicr	LinRegDoc.txl L-nTegVod.txl _irRegDetts: ion-LinRegr n-LinRegWintst t-bold stickyw/

Gambar 1.19. Tampilan GUI PBSModelling untuk Mencoba Berbagai Ilustrasi yang Disediakan



Gambar 1.20. Contoh Tampilan GUI PBSModelling untuk Regresi Linier dengan Ilustrasi Grafik yang Dihasilkan

I Made Tirta

1.9 Latihan

Untuk memeriksa pemahaman anda pada bab ini, jawab pertanyaan berikut tanpa melihat buku teks. Setelah semua jawaban anda tulis lalu cocokkan jawaban anda dengan dengan ide utama yang diuraikan pada bab ini. Pembaca tidak perlu mengutip kata demi kata untuk menunjukkan pemahaman akan meteri ini.

- 1. Apa yang dimaksud dengan program open source?
- 2. Apa yang anda ketahui tentang program R?
- 3. Apa keuntungan menggunakan program R?
- 4. Kunjungi situs R, catat beberapa referensi (buku teks atau jurnal) yang khusus membahas R ataupun menggunakan R sebagai alat analisis.

2006

2 KOMPONEN R-GUI

Pembaca yang sering menggunakan program komputer, pasti telah biasa melihat tampilan menu berbasis grafis dari suatu program komputer. Tampilan menu ini sering juga disebut sebagai GUI (*Graphical User Interface*). Pada bab ini akan dibahas komponen GUI untuk R terutama yang dihasilkan dengan pustaka R-TclTk, yaitu tclk dan tcltk2. Pada akhir pembahasan bab ini pembaca diharapkan memahami bermacam-macam jendela yang dapat dibuat oleh R-TclTk dan komponen-komponen widget lainnya yang dapat dipasang pada suatu jendela; serta dapat meletakkan komponen widget dalam suatu jendela

2.1 Widget dengan Elemennya

GUI (*Grafphicall User Interface*) merupakan *interface* (antarmuka) yang bersifat grafis yang menjembatani antara pengguna program (*user*) dan program. Dengan pendekatan GUI pengguna umumnya tidak perlu memiliki kemampuan bahasa yang dipakai dalam program tapi cukup dengan memilih menu yang tersedia baik dengan menggunakan keyboard maupun mouse. Komponen GUI disebut Widget (Dalgaard 2001). Widget untuk R-GUI, didefinisikan melalui paket yang sejauh ini ada dua pustaka yaitu tcltk (Dalgaard, 2001; 2003) dan tcltk2 (Grosjean 2005; 2006). Widget terdiri atas beberapa elemen mendasar diantaranya seperti berikut ini.

1. **Jendela**. Ada dua jenis jendela dalam R-GUI yaitu yang pertama adalah jendela dialog (*dialog window*) yang merupakan jendela utama. Jendela ini dapat memuat berbagai elemen widget lainnya. Yang ke dua adalah jendela pesan (*message window/box*) yang

I Made Tirta

biasanya hanya berisi pesan singkat berupa informasi atau peringatan. Jendela pesan ini dilengkapi *icon/*ikon sesuai jenis pesannya (diantaranya *warning, info,* dan *error*)

2. Elemen widget lain seperti menu, bingkai/frame, slider, teks/label, tombol dan kanvas.

Berikut adalah beberapa contoh tampilan jendela pesan dan jendela dialog dengan beberapa elemennya

An error has occurred!	This is a warning!
ОК	<u>OK</u>

Gambar 2.1. Jendela Pesan dengan Ikon *Error* dan *Warning*. Pemilihan ikon disesuaikan dengan jenis pesan yang disampaikan.

2006


Gambar 2.1 Jendela Dialog dengan Beberapa Elemen seperti Kotak Daftar, Kotak Edit dan Berbagai Tombol.

I Made Tirta



Gambar 2.2. Jendela Dialog dengan Frame, Slider dan Beberapa Tombol

2.2 Menempatkan Elemen Widget

Penempatan elemen widget diistilahkan dengan **manajer geometri**. Penempatan posisi ini dilakukan dengan perintah tertentu dengan kode penempatan arah yang disebut *sticky* atau *anchor*. Opsi yang tersedia untuk penempatan posisi adalah

- 1. n (north), untuk posisi atas;
- 2. e, (east), untuk posisi kanan;
- 3. s, (*south*), untuk posisi bawah:
- 4. w, (west), untuk posisi kiri;
- 5. ne, untuk posisi pojok kanan atas;
- 6. ns, untuk tengah-tengah secara vertikal;
- 7. es, untuk posisi pojok kanan bawah;
- 8. ws, untuk posisi pojok kiri bawah;
- 9. wn, untuk posisi pojjok kiri atas;
- 10. we, untuk posisi tengah-tengah secara horisontal;
- 11. news, untuk posisi tengah-tengah secara keseluruhan.





I Made Tirta

2.3 Bacaan Lebih Lanjut

Untuk lebih mendalami pengetahuan tentang Widget dan TclTk, pembaca dapat membaca referensi umum yaitu Welch (1995). Referensi lebih spesifik untuk R dapat dilihat pada Dalgaard (2001) dan Dalgaard (2002).

2.4 Latihan

- 1. Sebutkan dua macam jendela yang dapat dibuat oleh R-TclTk
- 2. Sebutkan elemen-elemen widget yang dapat dipasang pada suatu jendela;
- 3. sebutkan cara meletakkan elemen widget dalam suatu jendela

3 DESAIN WIDGET SEDERHANA

Setelah kita mempelajari teori umum mengenai widget, pada bab ini pembaca diajak mendesain widget sederhana dengan RTcITk. Setelah membaca materi pada bab ini pembaca diharapkan dapat membuat dan memberi nama jendela serta menulis label pada jendela yang dibuat dengan berbagai jenis dan ukuran fon dan mengatur posisinya dalam jendela

3.1 Membuat Jendela sederhana

Jendela sederhana hanya memiliki beberapa elemen seperti nama/judul, teks atau label. TclTk telah dilengkapi fungsi-fungsi khusus untuk membuat jendela maupun elemennya. Untuk dapat memulai mendesain widget kita harus memanggil pustaka tcltk dengan perintah

library(tcltk) atau

require(tcltk)

Untuk menghindarkan pemanggilan berulang-ulang terhadap pustaka yang sama, pada bagian awal program ditulis perintah ke dua (require()), dengan cara ini R memeriksa apakah pustaka yang dimaksud sudah aktif atau belum dan memanggil jika belum aktif.

Untuk membuat jendela dialog kita gunakan fungsi tktoplevel() sedangkan untuk membuat jendela pesan dengan tkmessageBox (message="Pesan")

I Made Tirta

Untuk membuat jendela dialog, kita harus memanggil pustaka tcltk dan membuat objek jendela, seperi berikut ini.

require(tcltk)

ju<-tktoplevel()

Perintah di atas akan menghasilkan jendela pada Gambar 3.1 (bagian kiri) yang berupa jendela dialog yang masih kosong dengan nomor 1 (artinya jendela pertama yang dibuat tcltk pada saat periode aktif tersebut).



Gambar 3.1. Tampilan Jendela Dialog Sebelum Diberi Nama.

Selanjutnya jendela perlu diberi nama dan teks sebagai dialog. Untuk memberi nama jendela digunakan

tkwm.title(jendela, "Nama Jendela")

2006

atau tktitle(jendela)<-"Nama Jendela".

Apabila salah satu perintah di atas ditambahkan pada sekrip sebelumnya maka kita memperoleh perubahan tampilan jendela seperti Gambar 3.1 bagian kanan.

%Nama Jendela	- 🗆 🗙

Gambar 3.2. Tampilan Jendela Dialog Sesudah Diberi Nama.

Selanjutnya untuk mendefinisikan label dilakukan dengan fungsi tklabel() dan penempatan label dilakukan dengan fungsi tkgrid() dengan sintak seperti berikut.

ObjekLabel<-tklabel(jendela, text="Label/ teks tulisan") tkgrid(ObjekLabel, sticky=(n,e,s,w))

Misalkan kita membuat objek label dan penempatannya sebagai berikut: Teks kiri ditempatkan dengan posisi "w", Teks kanan dengan posisi "e" yang lainnya tanpa diberi arah posisi. Hasil tampilan ditunjukkan oleh Gambar 2.

I Made Tirta

```
tkgrid(tklabel(ju,text="Ini adalah Teks Normal"))
lb1<-tklabel(ju,text="Teks kiri")
tkgrid(lb1,sticky="w")
lb2<-tklabel(ju,text="Teks kanan")
tkgrid(lb2,sticky="e")</pre>
```



Gambar 3.3.Tampilan Jendela Sederhana dengan Teks atau Label. Penutupan jendela dapat dilakukan dengan mengklik tanda (×) pada pojok atas jendela.

Ukuran dan Jenis fon untuk label dapat didefinisikan baik jenis maupun ukurannya denmngan fungsi tkfont.create(). Format fungsi untuk mendefinisikan fon adalah

Nama<-tkfont.create(family="",size=,weight="",slant="")

2006

Opsi untuk **family** yang tersedia diantaranya adalah **times, courier, sans**. Opsi untuk **slant** yang tersedia adalah **italic, roman** dan opsi untuk **weight** diantaranya adalah **bold**.

```
require(tcltk)
ju <- tktoplevel()</pre>
tkwm.title(ju,"Contoh Fon")
TeksJudul <- tkfont.create(family="sans", size=18,</pre>
             weight="bold",slant="roman")
TeksSub < tkfont.create(family="sans",</pre>
           size=16,slant="italic")
TeksNormal <- tkfont.create(family="times",</pre>
           size=12)
TeksSkrip <- tkfont.create(family="courier",</pre>
           size=12)
tkgrid(tklabel(ju,text="Fon Judul (Sans,18, bold,
           roman)", font=TeksJudul))
tkgrid(tklabel(ju,text="
                            "))
tkgrid(tklabel(ju,text="Fon Sub
Judul(Sans,16,italic)",font=TeksSub))
tkgrid(tklabel(ju,text="
                             "))
tkgrid(tklabel(ju,text="Fon Normal
               (Times, 12)", font=TeksNormal))
tkgrid(tklabel(ju,text=""))
tkgrid(tklabel(ju,text="Fon Skrip (Courier,
       12)",font=TeksSkrip))
tkfocus(ju)
```

I Made Tirta



Gambar 3.4. Tampilan Jendela dengan Berbagai Ukuran Label dan Jenis Fon.

3.2 Jendela dengan Tombol

Selanjutnya jendela dapat dilengkapi tombol tertentu, misalnya yang paling sederhana adalah tombol untuk menutup jendela. Pembuatan objek tombol , menghapus jendela dilakukan dengan fungsi

tkbutton(jendela,"LabelTombol",command=Fungsi)

Fungsi dapat didefinisikan dengan dua cara. Pertama, jika fungsinya pendek dapat langsung didefinisikan pada baris yang sama. Kedua, jika fungsinya panjang, maka perlu didefinisikan tersendiri. Berbeda dengan fungsi R untuk matematika dan statistika, fungsi-fungsi TclTk tidak memerlukan parameter. Fungsi yang paling sederhana yang bisa dibuat adalah fungsi yang menutup (*destroy*) jendela melalui perintah tkdestroy(NamaJendela).

Dengan menambahkan skrip berikut pada skrip sebelumnya maka diperoleh tampilan seperti pada Gambar 3.5. Jendela akan ditutup apabila tombol Tutup di-klik.



Gambar 3.5. Tampilan Jendela dengan Label dan Tombol Penutup Jendela.

3.3 Membuat Jendela Pesan

Ada kalanya diperlukan jendela untuk mengirim pesan tertentu sebagai reaksi dari pemilihan sebuah tombol, misalnya bisa berupa informasi, peringatan, pemberitahuan adanya kesalahan atau pertanyaan yang harus direspon pengguna. Untuk R melalui Tcltk perbedaan tampilan jendela pesan satu dengan lainnya ditentukan oleh pemilihan ikon yang telah tersedia sesuai jenis pesannya. Berikut adalah contoh pemanggilan jendela pesan dengan pemilihan pesan dan ikonnya.

tkmessageBox(message="pesan", icon=jenis.ikon)

I Made Tirta

dengan jenis ikon: "**info**", "**question**", "**warning**" dan "**error**". Berikut adalah contoh skrip pemanggilan jendela pesan beserta tampilan jendela yang dihasilkan.



Gambar 3.6 Contoh Tampilan Jendela Pesan dengan Pesan Informasi

2006



Gambar 3.7 Contoh Tampilan Jendela Pesan dengan Pesan Berupa Peringatan



Gambar 3.8. Contoh Tampilan Jendela Pesan dengan Pesan Berupa Pertanyaan.

I Made Tirta



Gambar 3.9. Contoh Tampilan Jendela Pesan dengan Pesan Berupa Pemberitahuan Kesalahan.

Adakalanya kita perlu menyediakan lebih dari satu pilihan, misalnya tombol Tutup (Batal) atau OK (Teruskan) untuk itu kita dapat mendefinisikan dua tombol dan mendefinisikan fungsi yang bergantung pada pilihan pengguna.

2006



Gambar 3.10. Tampilan Jendela dengan Tombol OK dan Batal



Gambar 3.11. Tampilan Jendela dengan Pesan yang Dihasilkan dari Pemilihan

I Made Tirta

3.3.1 Mengganti Label

Selanjutnya jika ingin mengganti label, kita dapat mendefinisikan fungsi dan tombol untuk mengganti label, misalnya fungsinya diberi nama GantiLabel dan tombolnya diberi nama label.but. Untuk mengakomodasi perubahan label, pada skrip diberi perintah

tkconfigure(objeklabel,textvariable=variabel)

```
require(tcltk)
ju <- tktoplevel()
tkwm.title(ju,"Nama Jendela")
TeksLabel <- tclVar("Label Lama")
lbl <- tklabel(ju,text=tclvalue(TeksLabel))
tkconfigure(lbl,textvariable=TeksLabel)
tkgrid(lbl)
GantiLabel <- function() tclvalue(TeksLabel) <-
        "Label Baru"
lbl.but <- tkbutton(ju,text="Ganti
label",command=GantiLabel)
tkgrid(lbl.but)</pre>
```

Skrip di atas masing-masingmenghasilkan tampilan seperti gambar berikut (sebelum dan sesudah tombol ganti label di-klik).

2006



Gambar 3.12. Jendela dengan Tombol Penggantian Label.

<mark>7% Nama Jendela</mark>	<u>- 🗆 ×</u>
Label Baru	
Ganti label	
19 10 - 1 91	

Gambar 3.13. Jendela dengan Tombol Penggantian Label dan Tampilan Sesudah Penggantian Label.

I Made Tirta

3.3.2 Menambah Kotak Cek dan Tombol Radio

Pilihan dalam jendela widget dapat disajikan dengan beberapa cara diantaranya melalui kotak cek (*check box*) atau tombol radio (*radio button*)

```
require(tcltk)
ju <- tktoplevel()
cb <- tkcheckbutton(ju)</pre>
tkwm.title(ju,"Contoh Kotak Cek")
cbValue <- tclVar("0")</pre>
tkconfigure(cb,variable=cbValue)
tkgrid(tklabel(ju,text="Klik Cek Jika
       Setuju"),cb)
OnOK <- function()
{
    cbVal <- as.character(tclvalue(cbValue))</pre>
    tkdestroy(ju)
    if (cbVal=="1")
     tkmessageBox(message="Anda Setuju!")
    if (cbVal=="0")
     tkmessageBox(message="Anda belum memberi
         respon!",icon="warning")
}
OK.but <- tkbutton(ju,text="OK",command=OnOK)
tkgrid(OK.but)
tkfocus(ju)
```

2006



Gambar 3.14. Tampilan Jendela dengan Kotak Cek dan Jendela Pesan ketika Kotak Cek tidak Di-klik.

I Made Tirta



Gambar 3.15. Tampilan Jendela dengan Kotak Cek dan Jendela Pesan Ketika Kotak Cek Di-klik.

Penggunaan tombol radio melalui tkradiobutton(), mirip dengan kotak cek hanya harus ada beberapa opsi dan pilihan berupa string atau karakter. Dalam skrip berikut pengguna diarahkan untuk menyetujui apa yang ditawarkan penulis. Pesan terimaksih muncul ketika pilihan setuju, tetapi sebaliknya pengguna disuruh berfikir ulang letika tidak setuju.

2006

```
OnOK <- function()
{
    rbVal <- as.character(tclvalue(rbValue))
    tkdestroy(ju)
    if (rbVal=="Setuju")
    tkmessageBox(message="Terimakasih Atas
    Dukungannya")
    if (rbVal=="Tidak Setuju")
    tkmessageBox(message="Pikirkan Kembali
    Pilihan Anda")
}
OK.but <- tkbutton(ju,text="OK",command=OnOK)
tkgrid(OK.but)
tkfocus(ju)</pre>
```

I Made Tirta



Gambar 3.16. Tampilan Jendela dengan Tombol Radio dan Jendela Pesan dengan Pilihan Setuju.

3.4 Jendela dengan Kotak Daftar

Selain dengan cara seperti diuraikan sebelumnya, tawaran pilihan dapat juga diberikan dalam bentuk kotak daftar.

• Kotak daftar dipanggil menggunakan fungsi

2006

tklistbox(jendela,height=,selectmode="single",background="white")

• Isian daftar diisi dengan

tkinsert(objekdaftar,"end",item)

Default pilihan ditentukan dengan memilih nomor item (yang mulai dari 0).

```
tkselektion(objekdaftar,no.item)
```

Berikut adalah contoh skrip yang menggunakan kotak daftrar dan tampilan yang dihasilkan.

```
require(tcltk)
ju<-tktoplevel()
tkwm.title(ju, "Kotak Daftar")
tl<-tklistbox(ju,height=3,selectmode="single",</pre>
     background="white")
tkgrid(tklabel(ju,text="Posisi Anda?"))
tkgrid(tl)
daftar <- c("Setuju", "Netral", "Tidak Setuju")</pre>
for (i in (1:3))
{
    tkinsert(tl,"end",daftar[i])
}
tkselection.set(tl,0) # Default setuju.
OK.but <-tkbutton(ju,text=" OK
                                     ", command=
        function()tkdestroy())
tkgrid(OK.but)
tkfocus(ju)
```

I Made Tirta



Gambar 3.17. Contoh Tampilan Kotak Daftar

3.5 Jendela dengan Kotak Isian

Selain memilih pilihan yang telah tersedia, pengguna dapat juga menulis sesuatu melalui kotak isian yang didefinisikan dengan fungsi tkentry(). Kita dapat memberikan isian awal yang berguna jika pengguna malas memasukkan isian.

```
require(tcltk)
ju<-tktoplevel()
tkwm.title(ju,"Jendela Isian")
Name <- tclVar("Tes")
entry.Name <-
tkentry(ju,width="20",textvariable=Name)
tkgrid(tklabel(ju,text="Tulis sesuatu"))
tkgrid(entry.Name)
OnOK <- function()
{</pre>
```

2006

```
NameVal <- tclvalue(Name)
    tkdestroy(ju)
    msg <- paste("Anda Menulis:",NameVal,"
        Terimaksih.")
    tkmessageBox(message=msg)
}
OK.but <-tkbutton(ju,text=" OK ",
        command=OnOK)
tkbind(entry.Name, "<Return>",OnOK)
tkgrid(OK.but)
tkfocus(ju)
```

🥻 Jendela Isian 💶 🗵		
Tulis sesuatu		
Tes		
OK		
Anda Menulis: Tes Terimaksih.		
ОК		

Gambar 3.18. Tampilan Jendela dengan Kotak Isian dan Jendela Pesan ketika Tombol OK Di-klik.

I Made Tirta

3.6 Jendela dengan Penggeser/ Slider

Pada pembahasan sebelumnya telah dijelaskan bahwa pemilihan suatu alternatif dapat dilakukan secara diskrit dengan berbagai cara (diantaranya tombol radio, kotak cek). Perubahann pilihan, terutama yang bersifat numerik, dapat juga dilakukan secara kontinu dengan menggunakan *slider*. Dengan slider perubahan variabel dapat dilakukan secara kontinu pada suatu interval tertentu.

Bentuk umum slider ini adalah

```
tkscale(jendela, from=, to= showvalue=T, variable=NamaVar,
```

resolution=1, orient="vertical")

dengan

- from ...to ...menunjukkan batas bawah dan batas atas interval nilai slider
- showvalue T/F mengontrol penampilan nilai (ditunjukkan/tidak)
- resolution mengatur besarnya perubahan pergeseran
- orient mengatur posisi slider pilihan vertical atau horizontal

Pada skrip berikut dibuat dua slider (tegak dan datar) masing-masing dengan interval dari 0 sampai dengan 100, kenaikan/resolusi 1, nilai dimunculkan, orientasi yang satu horizontal yang satu lagi vertikal.

```
require(tcltk)
ju <- tktoplevel()
tkwm.title(ju,"Contoh Slider")
SliderValue <- tclVar("50")
SliderValueLabel <- tklabel(ju,
        text=as.character(tclvalue(SliderValue)))
tkgrid(tklabel(ju,text="Nilai Slide : "),
        SliderValueLabel,tklabel(ju,text="%"))
tkconfigure(SliderValueLabel,textvariable=
        SliderValue)</pre>
```

Laboratorium Statistika-Jurusan Matematika FMIPA UNEJ



Gambar 3.19. Tampilan Silder Tegak dan Datar. Nilai variabel dapat diubah dengan menggeser slider keatas bawah atau ke kiri kanan.

I Made Tirta

3.7 Bacaan Lebih Lanjut

Sebagian besar teori yang dipelajari pada bab ini dapat dibaca pada Grosjean (2006).

3.8 Latihan

Sebagai bahan latihan, buat jendela-jendela berikut.

- 1. Jendela dengan judul: "Jendelaku", label bertuliskan: "Pesanku", dengan menggunakan font sans, miring, dengan ukuran 14.
- 2. Jendela seperti pada soal 1, tetapi dilengkapi dengan tombol penutup jendela.
- 3. Jendela dengan judul: "tombol radio" dengan tiga pilihan: "ya, raguragu, tidak".
- 4. Jendela geser dengan judul: "tombol geser vertikal", dengan nilai interval dari 10 sampai dengan 200, dengan kenaikan 1 dan orientasi vertikal.;

2006

4 DESAIN WIDGET KOMPLEKS

Pada bab ini kita akan belajar mendesain jendela yang lebih kompleks dengan menggabungkan elemen-elemen widget yang telah dipelajari sebelumnya. Pembaca diharapkan dapat membuat jendela dengan beberapa bingkai; jendela dengan menu dan submenu; jendela teks dan jendela skrip; jendela komunikasi dengan sistem (terutama untuk membaca dan menyimpan file), serta menggabungkan elemen-elemen widget dalam satu jendela dialog.

4.1 Jendela dengan Bingkai

Pembuatan bingkai (frame) dapat dilakukan dengan menggunakan fungsi berikut

tkframe(jendela, relief="", borderwidth=)

Pilihan untuk relief ada beberapa opsi diantaranya: "**groove**", "**raised**", "**sunken**", "**flat**", "**ridge**" dan "**solid**". Penempatan bingkainya dapat dilakukan dengan menggunakan

```
tkpack(nama frame, fill="x")
tkpack(jenis frame(kiri,kanan) ,side="left", anchor="n")
```

Berikut adalah skrip dengan beberapa posisi bingkai (atas, bawah, kiri atau kanan). Di dalam masing-masing frame, dapat ditempatkan beberapa elemen widget lain (tombol radio, kotak cek, slider dan lain-lain) seperti yang telah dibahas sebelumnya.

I Made Tirta

```
require(tcltk)
base <- tktoplevel()</pre>
tkwm.title(base, "Contoh Frame")
#mendefinisikan jenis frame
spec.frm <- tkframe(base, borderwidth=2)</pre>
left.frm <- tkframe(spec.frm)</pre>
right.frm <- tkframe(spec.frm)</pre>
## Frame Atas
frame1 <- tkframe(base, relief="groove",</pre>
       borderwidth=2)
tkpack(tklabel(frame1, text=" Frame1 Atas Groove
"))
#Frame Kiri
frame2 <-tkframe(left.frm, relief="flat",</pre>
     borderwidth=2)
tkpack(tklabel(frame2, text=" Frame2 Kiri 1 Flat
"))
frame5 <-tkframe(left.frm, relief="solid",</pre>
     borderwidth=2)
tkpack(tklabel(frame5, text=" Frame5 Kiri 2 Solid
"))
#Frame Kanan
frame3 <-tkframe(right.frm, relief="raised",</pre>
     borderwidth=2)
tkpack(tklabel(frame3, text=" Frame3 Kanan1
Raised "))
frame4 <-tkframe(right.frm, relief="sunken",</pre>
      borderwidth=2)
tkpack(tklabel(frame4, text=" Frame4 Kanan2
Sunken "))
#Menempatkan frame
tkpack(frame1, fill="x")
```

2006

```
tkpack(frame2, frame5, fill="x")
tkpack(frame3,frame4, fill="x")
tkpack(left.frm, right.frm,side="left",
anchor="n")
#Tombol lain
q.but <- tkbutton(base,text="Tutup",
command=function()tkdestroy(base))
tkpack(spec.frm, q.but)</pre>
```

% Contoh Frame	_ 🗆 ×
Frame1 Atas Groove	
Frame2 Kiri 1 Flat	Frame3 Kanan1 Raised
Frame5 Kiri 2 Solid	Frame4 Kanan2 Sunken
Tutup	

Gambar 4.1. Tampilan Jendela dengan Beberapa Bingkai dan Variasinya.

Untuk melatih ketrampilam membuat komponen R-GUI, salin sakrip yang ada lakukan beberapa modifikasi dan jalankan pada R (versi 1.9.1. ke atas). Buat Jendela dialog dengan nama sesuai yang anda kehendaki selanjutnya beri label dengan berbagai jenis dan ukuran fon, baik yang telah didefinisikan maupun yang anda definisikan sendiri serta beberapa komponen GUI lainnya. Jangan takut melakukan "trial & error". Semakin banyak anda berlatih semakin tertanam *sense/art of programming* terutama dengan menggunakan RTclTk.

4.2 Membuat Menu Tunggal

I Made Tirta

Menu kita anggap sederhana jika hanya terdiri atas satu tingkat pilihan. Untuk membuat menu kita menggunakan fungsi khusus dari TclTk diantaranya adalah:

NamaMenu<-tkmenu(jendela) tkadd(Nama.Menu, jenis, label="", command=fungsi)

Pilihan yang tersedia untuk parameter tkadd adalah

- jenis terdiri dua macam yaitu cascade untuk pilihan yang berupa submenu atau command untuk pilihan yang langsung mengeksekusi perintah.
- **label** adalah nama yang diberi untuk pilihan; nama yang diberikan ini akan muncul dalam tampilan.
- fungsi mendefinisikan perintah atau submenu yang dipanggil

Menu sederhana yang dimaksud dalam subbab ini hanya terdiri atas menu yang tidak memiliki submenu tetapi langsung memanggil perintah. Pada contoh berikut pilihan menu langsung berhubungan dengan fungsi command dalam hal ini semua diisi tkdestroy() yang fungsinya menutup jendela.

```
require(tcltk)
ju <- tktoplevel()
tkwm.title(ju,"Contoh Menu")
topMenu <- tkmenu(ju)
tkconfigure(ju,menu=topMenu)
Menul <- tkmenu(topMenu,tearoff=FALSE)
tkadd(Menul,"command",label="Pilihan1",command=
function() tkdestroy(ju))
tkadd(Menul,"command",label="Pilihan2",command=
function() tkdestroy(ju))
tkadd(Menul,"command",label="Pilihan3",command=
function() tkdestroy(ju))
```

tkadd(topMenu,"cascade",label="Menu1",menu=Menu1)

2006

tkadd(topMenu,"cascade",label="Menu2",menu=Menu1)
tkfocus(ju)



Gambar 4.2. Tampilan Menu Sederhana dengan Pilihan Tanpa Submenu

4.3 Membuat Menu Bertingkat

Menu yang telah dibuat sebelumnya dapat dikembangkan dengan memberikan pilihan submenu satu dua atau tiga tingkat sesuai keperluan. Pada dasarnya penambahan submenu dilakukan dengan beberapa langkah berikut ini.

- 1. Mendefinisikan nama-nama kelompok menu dan jendelanya
 - topMenu <- tkmenu(ju)
 - SubMenu <- tkmenu(ju)
 - SubSubMenu <- tkmenu(ju)
- 2. Mendefinisikan posisi masing-masing kelompok menu
 - menut<- tkmenu(topMenu,tearoff=FALSE)

I Made Tirta

- **SM<-**tkmenu(SubMenu,tearoff=FALSE)
- ssm<-tkmenu(SubSubMenu,tearoff=FALSE) Pilihan tearoff=FALSE artinya menu tidak bisa dilepas, sedangkan tearoff=TRUE artinya menu bisa dilepas
- Mendefinisikan submenu masing-masing kelompok menu, untuk submenu jenisnya adalah cascade dan perintahnya memanggil menu (yang tingkatnya lebih rendah)

```
tkadd(menut,"cascade",label="SubPilihan", menu=sm)
tkadd(sm,"cascade",label="SubPilihan", menu=ssm)
tkadd(sm,"command",label="SubPilihan", command=Fs)
```

Berikut adalah pengembangan skrip sebelumnya dengan sedikit memodifikasi label dan menambahkan beberapa tingkatan submenu. Pada contoh berikut Menu II dapat dilepas.

```
require(tcltk)
ju <- tktoplevel()
tkwm.title(ju,"Contoh Menu")
topMenu <- tkmenu(ju)</pre>
SubMenu <- tkmenu(ju)
SubSubMenu <- tkmenu(ju)
tkconfigure(ju,menu=topMenu)
Menu1 <- tkmenu(topMenu,tearoff=FALSE)</pre>
Menu2 <- tkmenu(topMenu,tearoff=TRUE)</pre>
sml<-tkmenu(SubMenu,tearoff=FALSE)</pre>
sm2<-tkmenu(SubSubMenu,tearoff=FALSE)</pre>
tkadd(Menu2, "command", label="Pilihan II 1",
command=function() tkdestroy(ju))
tkadd(Menu2,"command",label="Pilihan II 2",
command=function() tkdestroy(ju))
tkadd(Menul,"command",label="Pilihan I 1",
      command=function() tkdestroy(ju))
tkadd(Menul,"cascade",label="Pilihan I 2",
```

Laboratorium Statistika-Jurusan Matematika FMIPA UNEJ

```
menu=sm1)
tkadd(Menul,"command",label="Pilihan I 3",
    command=function() tkdestroy(ju))
tkadd(sm1,"command",label="Sub
    Pilihan1",command=function() tkdestroy(ju))
tkadd(sm2,"command",label="Sub-Sub
    Pilihan1",command=function() tkdestroy(ju))
tkadd(sm2,"command",label="Sub-Sub
    Pilihan1",command=function() tkdestroy(ju))
tkadd(sm1,"cascade",label="SubPilihan
    2",menu=sm2)
tkadd(topMenu,"cascade",label="MenuI",menu=Menu1)
```



Gambar 4.3. Tampilan Jendela dengan Menu yang Tidak Bisa Dilepas (Menu I) dan Menu yang Bisa Dilepas (Menu II)

I Made Tirta

4.4 Membuat Jendela Teks

Dalam bekerja dengan komputer kita membutuhkan jendela untuk menulis teks input, atau jendela yang menampung keluaran. Jendela ini diistilahkan dengan jendela teks. Jendela teks ini mungkin hanya bisa dibaca mungkin juga bisa diedit. Dari jendela input diharapkan skrip yang ditulis dapat langsung dikirim perintah ke program induk dalam hal ini R. Berikut adalah Jendela yang isinya bisa diedit. Fungsi penting untuk membuat jendela teks adalah

- tktext(jendela) : menentapkan objek teks
- tkgrid(txt) menyatukan objek teks dengan widget
- tkmark.set(txt,"insert","0.0") mengatur posisi awal kursor
- tkinsert(txt,"end","Anda Bisa Mengedit") mengatur tampilan teks awal
- tkconfigure(txt, state="disabled"), untuk mengatur agar jendela tidak bisa diedit.

Berikut adalah contoh jendela dengan tampilan teks yang bisa diedit.

```
require(tcltk)
je <- tktoplevel()
tkwm.title(je, "Jendela Edit")
txt <- tktext(je)
tkgrid(txt)
tkmark.set(txt,"insert","0.0")
tkinsert(txt,"end","Anda Bisa Mengedit")
tkfocus(txt)</pre>
```

2006


Gambar 4.4. Tampilan Jendela Edit dengan Teks yang Bisa Diedit. Jendela ini berguna untuk menulis skrip program

Untuk menonaktifkan kemampuan mengedit, maka pada skrip harus ditambahi dengan perintah tkconfigure(txt, state= "disabled"). Pada skrip berikut tampilan teks yang ada pada jendela hanya bisa dibaca, tetapi tidak bisa diedit.

```
require(tcltk)
jne <- tktoplevel()
tkwm.title(jne, "Jendela Baca")
txt <- tktext(jne)
tkgrid(txt)
tkmark.set(txt,"insert","0.0")
tkinsert(txt,"end"," Anda tidak bisa mengedit
teks dalam jendela ini.")
tkconfigure(txt, state="disabled")
tkfocus(txt)</pre>
```

I Made Tirta



Gambar 4.5. Tampilan Jendela dengan teks yang hanya bisa dibaca dan tidak bisa diedit.

4.5 Dialog dengan Sistem Operasi

Dalam bekerja dengan program komputer, kita sering perlu membuka dan menyimpan file. Untuk itu diperlukan dialog atau komunikasi antara program dengan sistem operasi. TclTk memiliki beberapa fungsi yang terkati dengan kebutuhan membuka dan menyimpan file yaitu:

- 1. fungsi tkgetOpenFile() fungsi untuk membuka file dan
- 2. fungsi tkgetSaveFile() untuk menyimpan file.

Berikut adalah contoh pemakaian kedua fungsi tersebut.

```
require(tcltk)
Nama.File<-tclvalue(tkgetOpenFile())
if (!nchar(fileName))
    tkmessageBox(message="Belum Memilih File")
else
    tkmessageBox(message=paste("File yang
    diambil",Nama.File))</pre>
```

2006

Cluster ☐ gam ☐ Intest ☐ mvnormtest CoCoAn ☐ gee ☐ maptools ☐ mvtnorm	ade4 amap base car car class cluster CoCoAn	dynamicGraph dynamicGraph effects ff Extremes fields foptions foreign gam gee	graphics grDevices grid ismev KernSmooth lattice Intest maptools	Mass mathgraph MCMCpack methods mgcv multcomp mvnormtest mvtnorm	
--	--	---	---	---	--



Untuk membaca file dapat dilakukan dengan skrip berikut.

```
require(tcltk)
Nama.file<-tclvalue(tkgetSaveFile())
if (!nchar(fileName))
    tkmessageBox(message="Tidak ada nama yang
    dipilih",icon="warning")
else
    tkmessageBox(message=paste("File disimpan
    dengan nama",Nama.file),icon="info")</pre>
```

Tampilan yang dihasilkan adalah seperti pada Gambar berikut.

I Made Tirta

afm	Src) Faq जिल्लान	Thanks
 Ddoc	BDataTmp	Mus News	unins000.u
🗋 etc	Rhistory	Readme	Y2k
🗋 include	Authors A	README.packages	
🗅 lib	🛋 Changes	README.Rterm	
library	Copying	README.rw2000	
modules		RESOURCES	
] share		m -FAU	

Gambar 4.7. Tampilan Jendela Dialog untuk Menyimpan File.

Apabila nama file telah ditulis dan penyimpanan berhasil, maka akan muncul pesan seperti berikut.

-	×
٩	File disimpan dengan nama C:/Program Files/R/rw2000/tes
	ΟΚ

Gambar 4.8. Tampilan Komentar Setelah Menyimpan File.

4.6 Widget dengan tcltk2

າດ	<u>م</u>
20	00

Tahun 2005 Grosjean menulis pustaka tcltk2 yang dapat melengkapi kebutuhan untuk membuat R-GUI. Diharapkan dengan tcltk2 desain R-GUI menjadi lebih mudah. Namun, pustaka tcltk2 ini masih dalam proses penyelesaian sehingga belum banyak contoh aplikasi yang diberikan.

4.6.1 Membuat Tips pada widget

Paket tcltk2 menyediakan fasilitas tooltip untuk membuat tips (petunjuk singkat) yang muncul secara otomatis ketika mouse digerakkan/ditempatkan diatas bagian yang ditentukan. Berikut adalah struktur untuk membuat tips dalam menu yang disebut tooltip.

```
tk2tip(nama_widget, "Label")
```

"Label" adalah teks/tips yang akan muncul pada saat mouse digerakkan pada wilayah (bagian) yang ditentukan. Selain itu seperti widget pada umumnya petunjuk menggunakan tips dapat dimunculkan dengan menggunakan tk2label. Contoh berikut dimodifikasi dari Grosjean (2005).

I Made Tirta

🎀 Tips		L D X dua
Gerakkan mouse di ata	s saya, atau di ata	s tomb <u>ol untuk melihat tip</u> in.
	ок	Contoh tip dengan label dalam dua baris

Gambar 4.9. Contoh Tampilan Jendela dengan Tips

Pada contoh berikut jendela widget mempunyai 4 wilayah dengan empat macam tips 2 berupa teks dan 2 lagi berupa tombol.

```
library(tcltk)
library(tcltk2)
tt2 <- tktoplevel()</pre>
tkwm.title(tt2, "Tips")
lb1 <- tk2label(tt2, text = "Wilayah 1")</pre>
tkgrid(lb1)
lb2 <- tk2label(tt2, text = "Wilayah 2")</pre>
tk2tip(lb1, "Contoh tip 1")
tkgrid(lb2)
tk2tip(lb2, "Contoh tip 2")
but1 <- tk2button(tt2, text = "Tutup", width = 10,</pre>
        command = function() tkdestroy(tt2))
tkgrid(but1)
tk2tip(but1, "Ini akan menutup jendela")
but2 <- tk2button(tt2, text = "OK", width = 10,</pre>
        command = function() tkfokus(tt2))
tkgrid(but2)
tk2tip(but2, "Ini tidak berbuat apa-apa")
```

2006



Gambar 4.10. Contoh Tampilan dengan Beberapa Tips. Jika mouse digerakkan pada wilayah 1, wilayah 2, tombol Tutup dan OK akan muncul tips yang berbeda yang menjelaskan apa yang akan terjadi jika mouse di-klik.

4.6.2 Variasi Tampilan Jendela dengan tcltk2

Paket tcltk2 juga menawarkan beberapa alternatif tampilan jendela yang sedikit lebih kompleks dari jendela-jendela yang telah dibahas sebelumnya. Dua tampilan jendela yang cukup menarik adalah *notebook*, dengan tk2notebook(), yang dapat digunakan sebagai alternatif menampilkan menu dan *panedwindow*, dengan tk2panedwindow() sebagai alternatif untuk menampilkan frame.

tk2notebook(jendela) tk2panedwindow(jendela, orient = "")

Berikut adalah contoh penggunaan *notebook* dengan dua pilihan menu (menuframe 1, dan menu frame 2) dan masing-masing menu dapat menampilkan pilihan yang berbeda. Pada contoh berikut salah menuframe1 tanpa ada pilihan sedangkan menuframe2 memiliki tombol untuk tutup jendela.

I Made Tirta



Gambar 4.11. Contoh Tampilan Jendela Notebook dengan tcltk2 dengan Dua Frame. Pengaktifan salah satu bingkai akan menampilkan pilihan yang berbeda (tidak ada tombol atau tombol tutup)

Panedwindow digunakan untuk membagi jendela menjadi beberapa bagian baik secara horizontak maupun vertikal yang dinyatakan dalam pilihan orientasi. Pilihan orientasi ada dua yaitu orient= "**vertical**" untuk pembagian secara vertikal atau orient= "**horizontal**" untuk pembagian secara horisontal. Pada jendela yang ditampilkan, pembagian dapat diubah secara dinamik dengan menggnakan mouse. Berikut adalah pembagian jendela baik secara vertikal maupun horizontal.

2006

```
tt2 <- tktoplevel()
tkwm.title(tt2,"TclTk2:Panedwindow")
pw <- tk2panedwindow(tt2, orient = "horizontal")
lpw.1 <- tk2label(pw, text = "Panel 1")
lpw.2 <- tk2label(pw, text = "Panel 2")
tkadd(pw, lpw.1, minsize = 100)
tkadd(pw, lpw.2, minsize = 70)
tkpack(pw, fill = "both", expand = "yes")</pre>
```

74 TclTk2:Panedwi	ndow 🔳 🗖 🔀	74 T 🗆 🛛 🛛
Panel 1	Panel 2	Panel 1
		Panel 2

Gambar 4.12. Contoh Tampilan Jendela Panel Window dengan tcltk2 dengan Panel Berorietnasi Horizontal dan Vertikal.

I Made Tirta

4.7 Bacaan Lebih Lanjut

Sebagian besar teori yang dipelajari pada bab ini dapat dibaca pada Grosjean (2006), Zhang (2007)

4.8 Latihan

- 1. Salin transkrip program yang ada pada Lampiran buku ini untuk menghasilkan widget jendelal skrip, dilengkapi dengan menu edit (salin, tempel, hapus) dengan memberi judul sesuai selera anda
- 2. Dengan memodifikasi program yang telah diajarkan, buat skrip yang dapa menghasilkan widget berikut

2006

Interval I	Keyakinan
C 90%	
• 95%	
C 99%	
Banyaknya Simulasi	NSampel
C 1	40
C 10	
C 50	StDev
• 100	5
C 200	
C 500	
C 1000	
C 5000	

I Made Tirta

5 APLIKASI R-GUI

Tujuan Umum

Setelah memahami cara membuat widget baik sederhana maupun kompleks, pada bab ini, kita akan mempelajari cara menghubungkan elemen-elemen widget yang telah dipelajari dengan fungsi atau perintah R. Pembaca diharapkan dapat membuat jendela skrip dengan menu sederhana; membuat jendela dialog dengan TkRplot beserta perintah atau fungsi R yang terkait.

5.1 Membuat Jendela Dialog

Dalam membuat interface grafis, kita sering memerlukan jendela dialog dengan pilihan dengan beberapa pilihan misalnya

- 1. kotak isian yang memberi masukan informasi nama atau sejenisnya;
- 2. beberapa tombol pengambilan keputusan misalnya Ya/terus, Tidak/Batal dan tombol Bantuan yang biasanya memanggil informasi cara mengisi dialog.

Karena pemakaiannya yang sering berulang-ulang maka jendela ini dapat dibuat dalam bentuk fungsi dengan atau tanpa parameter. Berikut adalah contoh fungsi yang dimodifikasi dari RCmdr oleh Fox (2005).

2006

5.2 Jendela Editor Skrip Sederhana

Sebagaimana telah dibahas sebelumnya bahwa dengan TclTk kita bisa membuat jendela yang bisa di edit. Langkah berikutnya yang diperlukan adalah kemampuan untuk mengirim teks yang ada pada jendela ke R-Console. Untuk keperluan ini selain kemampuan membuat jendela editor yang telah dibicarakan didepan, diperlukan juga fungsi yang akan menerjemahkan teks pada jendela menjadi perintah yang diterima R-Console. Fungsi ini adalah:

- 1. tkget() adalah fungsi untuk menyalin teks yang ada pada jendela skrip;
- 2. **parse()** adalah fungsi yang menyalin teks dalam jendela menjadi perintah pada R-Console;
- 3. eval() adalah fungsi untuk mengevaluasi teks yang ada pada jendela editor;

Berikut adalah jendela skrip sederhana, yang hanya berisi jenbdela skrip dan tombol untuk mengirim perintah. Pada jendela kita dapat menulis berbagai perintah R dan mengirimnya melalui tombol kirim. Bila pada jendela ada perintah untuk menggambar grafik,maka jendela grafik akan langsung dibuka, sedangkan hasil cetakan keluaran akan ditulis pada R-Console.

```
require(tcltk)
ju <- tktoplevel()
tkwm.title(ju,"Skrip Editor")
tkpack(teks <- tktext(ju))
tkinsert(teks, "0.0", "plot(1:10)")
eval.txt <- function()
    eval(parse(text=tclvalue(tkget(teks, "0.0",
        "end"))))</pre>
```

I Made Tirta





Gambar 5.1. Tampilan Jendela Skrip (a) dan Grafik yang Dihasilkan dari Pengiriman Perintah yang Ada pada Jendela Skrip (b)

2006

5.3 Jendela Skrip dengan Menu

Selanjutnya penampilan jendela skrip dapat ditingkatkan dengan menambah menu. Menu yang paling sederhana berisi pilihan membuka atau menyimpan file serta mengirim (mengeksekusi) teks yang ada pada jendela skrip. Hal ini dapat dilakukan dengan menggabungkan elemenelemen widget yang telah dibahas sebelumnya. Skrip berikut merupakan gabungan dari skrip sebelumnya ditambah. Skrip editor yang lebih lengkap, yang mampu mengantisipasi kesalahan file, dapat dilihat pada Daalgard (2002). Jendela yang dihasilkan telah mampu membaca skrip dari file yang telah ada, menyimpan skrip pada jendela serta menjalankan skrip yang ada pada jendela.

I Made Tirta

```
tkadd(topMenu,"cascade",label="File",menu=Menu1)
tkadd(topMenu,"command",label="Kirim",
      command=eval.txt)
buka.file<-function() {</pre>
  file <- tclvalue(tkgetOpenFile())</pre>
  chn <- tkopen(file, "r")</pre>
  tkinsert(teks, "0.0", tclvalue(tkread(chn)))
  #tkclose(chn)
  wfile <<- file
}
 simpan.file<-function() {</pre>
    file <- tclvalue(tkgetSaveFile(</pre>
      initialfile=tclvalue(tkfile.tail(wfile)),
      initialdir=tclvalue(tkfile.dir(wfile))))
  if (!length(file)) return()
  chn <- tkopen(file, "w")</pre>
  tkputs(chn, tclvalue(tkget(txt,"0.0","end")))
  tkclose(chn)
  wfile <<- file
}
tkpack(teks)
```

2006

```
eval.txt <- function()
    eval(parse(text=tclvalue(tkget(teks, "0.0",
        "end"))))</pre>
```

% Skrip Editor	<u> </u>
File Kirim	
Buka Skrip	
Simpan Skrip	
Keluar	
y<-2*x+rnorm(n,0,2) z<-2*x+rnorm(n,0,2) print(x) pairs(x,y,z) plot(x,y)	

Gambar 5.2. Tampilan Jendela Skrip Editor yang Dapat Dipakai Mengedit dan Mengirim Perintah ke R.

Menu yang lebih kompleks, yang lebih menyerupai sebuah paket program dapat ditambahkan berupa menu untuk memanggil bantuan dokumentasi R. Untuk itu perlu dilakukan modifikasi mengganti menu kirim/eksekusi dengan menu bantuan, membuat definisi fungsi yang memanggil bantuan serta memindahkan tombol kirim/eksekusi ke bagian bawah jendela. Misalkan format bantuan yang diminta adalah format html, maka kita dapat melakukan perubahan pada skrip dengan.

I Made Tirta



Gambar 5.3 Tampilan Paket Editor R Sederhana. Jendela ini mampu mencatat, mengedit dan mengirim skrip serta dilengkapi menu memanggil bantuan R.

2006

Ketika menu Bantuan di klik maka akan muncul Bantuan format html seperti gambar berikut.



Gambar 5.4 Tampilan Jendela Bantuan R dengan Format html.

5.4 Editor Skrip dengan Menu Edit

Tuntutan akan editor yang lebih baik semakin meningkat. Tidak lengkap rasanya jika jendela editor tidak dilengkapi fasilitas mengedit seperti *copy* (salin), *paste* (tempel), *cut* (potong) dan *delete* (hapus). Untuk keperluan ini kita bisa mendefinisikan menu tambahan beserta fungsinya. Dengan semakin banyaknya menu, kita bisa mengatur ulang penampilan menu, misalnya untuk memudahan menghapus jendela. menu keluar diletakkan di menu utama.

I Made Tirta

Skrip tambahan untuk menampilkan menu pengeditan, dimodifikasi dari skrip RCmdr oleh Fox adalah sebagai berikut.

```
tkadd(topMenu,"command",label="Keluar",command=functio
n() tkdestroy(ju))
Menu2 <- tkmenu(topMenu,tearoff=FALSE)</pre>
tkadd(Menu2,"command",label="Salin",command=salin)
tkadd(Menu2, "command", label="Tempel", command=tempel)
tkadd(Menu2, "command", label="Potong", command=potong)
tkadd(Menu2, "command", label="Hapus", command=hapus)
tkadd(topMenu,"cascade",label="Edit",menu=Menu2)
salin <- function() {</pre>
          focused <- tkfocus()</pre>
          selection <- strsplit(tclvalue(tktag.ranges</pre>
          (focused, "sel")), " ")[[1]]
          if (is.na(selection[1])) return()
          textedt <- tclvalue(tkget(focused,</pre>
          selection[1], selection[2]))
          tkclipboard.clear()
          tkclipboard.append(textedt)
          }
hapus <- function() {</pre>
          focused <- tkfocus()</pre>
          selection <- strsplit(tclvalue(tktag.ranges</pre>
(focused, "sel")), " ")[[1]]
          if (is.na(selection[1])) return()
          tkdelete(focused, selection[1],
          selection[2])
          }
potong <- function() {</pre>
          salin()
          hapus()
          }
tempel <- function() {</pre>
          focused <- tkfocus()</pre>
```

```
2006
```

Tampilan jendela setelah dilakukan modifikasi dan penambahan menu dapat dilihat pada Gambar 5.5. Kita bisa melakukan penyalinan, penghapusan teks seperti layaknya mengedit naskah.

riie	Edit Bantua	an Keluar
print(rn	Salin	
	Tempel	
	Potong	
	Hapus	

Gambar 5.5. Contoh Tampilan Jendela Editor dengan Beberapa Menu Pilihan

5.5 Membuat Grafik Dinamik

Dengan memanfaatkan paket tcltk, tcltk2 dan tkrplot, kita dapat membuat berbagai grafik yang parameternya dapat diatur melalui jendela widget baik berupa tombol radio, pengeser ataupun yang lainnya. Pada dasarnya grafik ini adalah gabungan antara jendela widget (sebagai pengatur parameter fungsi) dan jendela grafik yang menampilkan grafik yang dihasilkan. Dengan kata lain jendela grafik menampilkan grafik yang diperitahkan dari jendela widget. Adapun langkah-langkah dalam membuat grafik adalah sebagai berikut.

I Made Tirta

 Menentukan elemen grafik dan data yang diperlukan. Hal pertama yang perlu ditetapkan sebelum mulai membuat program dengan tkrplot adalah menentukan jenis grafik yang akan dibuat dan elemen yang diperlukan. Sebagai contoh misalkan kita akan membuat grafik diagram pencar dari data dengan menggambar garis regresinya, baik berdasarkan parameter aslinya maupun hasil estimasi. Dengan demikian kita membutuhkan paling tidak tiga vektor data yaitu x, y sebagai data asli. Untuk menggambar garis regresi R menyediakan beberapa alternatif diantaranya adalah dengan menggunakan fungsi abline() dan lm()

```
plot(x,y,type='p', xlab="Sumbu X",ylab="Sumbu
Y",main="Judul Grafik")
ablines(lm(y~x))
ablines(a,b)
```

 Menentukan data elemen. Setelah elemen teridentifikasi, selanjutnya kita menentukan cara memperoleh parameter grafik dalam hal di atas sesungguhnya merupakan data yang diperlukan. Misalkan pada kasus di atas kita memperoleh x dan y dengan mensimulasikan data acak dengan kondisi tertentu sedangkan data y1 diperoleh dengan menggunakan perintah abline() dan fungsi lm() pada R.

```
x<-seq(30,50,length=N)
y<-a+b*x+rnorm(N,0,sigma)</pre>
```

Andaikan kita juga ingin membuat diagram pencar dari data yang selain berdistribusi normal, kita perlu menuliskan cara membangkitkan data dari distribusi tersebut, misalnya untuk distribusi Gamma, Poisson atau yang lainnya.

```
y<-3+2*x+rgamma(N, 1,1/sigma)
y<-3+2*x+rpois(N, sigma)</pre>
```

2006

- 3. **Menentukan parameter yang bisa diatur dan bentuk kontrol.** Pada kasus diatas misalkan parameter yang bisa diatur melalui jendela widget adalah sebagai berikut ini.
 - a. jenis distribusi melalui tombol radio
 - b. jumlah sampel N dengan tombol radio
 - c. besarnya deviasi baku dengan penggeser
 - Jendela widget umum dengan judul "Demo Regresi". *Lay out* bingkai adalah satu bingkai atas (untuk distribusi) dan dua bingkai bawah (kiri untuk ukuran sampel dan kanan untuk deviasi baku dan interval keyakinan)

```
base <- tktoplevel()</pre>
tkwm.title(base, "Demo Regresi")
spec.frm <- tkframe(base,borderwidth=2)</pre>
left.frm <- tkframe(spec.frm)</pre>
right.frm <- tkframe(spec.frm)</pre>
#Satu bingkai
frame1 <- tkframe(base, relief="groove",</pre>
      borderwidth=2)
tkpack(tklabel(frame1, text="Distribusi"))
tkpack(tkradiobutton(frame1, command=regen,
        text="Normal", value=1, variable=dist),
        anchor="w")
tkpack(tkradiobutton(frame1, command=regen,
        text="T Normal", value=2, variable=dist),
        anchor="w")
}
## Dua bingkai:
frame3 <-tkframe(left.frm, relief="groove",</pre>
      borderwidth=2)
tkpack(tklabel(frame3, text="Ukuran Sample"))
for ( i in c(10, 30, 60, 100) ) {
  tmp <- tkradiobutton(frame3, command=regen,</pre>
```

I Made Tirta

```
text=i,value=i,variable=size)
    tkpack(tmp, anchor="w")
}
frame4 <-tkframe(right.frm, relief="groove",</pre>
        borderwidth=2)
tkpack(tklabel (frame4, text="StDev"))
tkpack(tkscale(frame4, command=regen, from=0.05,
      to=50.00, showvalue=T, variable=bw,
               resolution=0.05, orient="horiz"))
tkpack(tklabel (frame4, text="Interval
      Keyakinan"))
tkpack(tkscale(frame4, command=replot, from=0.90,
      to=0.995, showvalue=T, variable=ik,
               resolution=0.01, orient="horiz"))
tkpack(frame1, fill="x")
tkpack(frame3, frame4, fill="x")
tkpack(left.frm, right.frm,side="left",
       anchor="n")
## `Bottom frame' (on base):
q.but <- tkbutton(base,text="Selesai",</pre>
         command=function()tkdestroy(base))
tkpack(spec.frm, q.but)
regen()
})
}
```

 Menentukan komunikasi Widget dengan grafik. Menentukan komunikasi antara jendela widget dengan jendela grafik. Dalam hal ini termasuk mendefinisikan nilai awal serta konversi antara konstanta biasa ke objek widget dan sebalikntya.

```
y <- NULL
xlim <-NULL
```

2006

```
a<-10
b<-5
size <- tclVar(50)</pre>
dist <- tclVar(1)
     <- tclVar(1)
bw
     <-tclVar(0.975)
ik
bw.sav <- 1 # in case replot.maybe is called too</pre>
             early
sq<-1
replot <- function(...) {</pre>
    if (is.null(y)) return() # too early...
    p <- as.numeric(tclObj(ik))</pre>
    plot(x,y,type='p',col="red",main="Diagram
    Pencar & Sabuk Keyakinan",ylim=c(-10,100))
    abline(a,b,col="blue")
    abline(reg1, col="red")
    x1 < -seq(min(x), max(x), 0.1)
    sep < -qt(p, n-2) * sqrt(var(y)) * sqrt(1/n+(x1-
             mean(x)) * *2/sum((x-mean(x)) * *2))
    sa<-cf[1]+cf[2]*x1 + sep</pre>
    sb<-cf[1]+cf[2]*x1 - sep
    lines(x1, sa, col="red", lty=2)
    lines(x1, sb, col="red", lty=2)
    lines(x,0*x,col="green")
    ya<-seq(0,max(y),0.1)</pre>
    lines(0*ya,ya,col="green")
}
replot.maybe <- function(...)</pre>
{
    if (as.numeric(tclObj(bw)) != bw.sav) replot()
}
regen <- function(...) {</pre>
    n<<-as.numeric(tclObj(size))</pre>
    sg<-as.numeric(tclObj(bw))</pre>
    x << -seq(0, 15, length=n)
    mu<-a+b*x
```

I Made Tirta

```
if (tclvalue(dist) == "1") {
    y<<-rnorm(n,mu,sg)
    else {y<<-rnorm(n,mu,sg*sqrt(abs(x)))}
    #xlim <<- range(y) + c(-2,2)
    replot()
}</pre>
```

Dalam desain di atas pada widget ditampilkan

- tombol radio untuk pilihan distribusi: normal tidak normal
- tombol radio untuk pilihan ukuran sampel (10, 30, 60, 100)
- slider (penggeser) untuk deviasi baku
- slider (penggeser) untuk nterval keyakinan

Sedangkan pada grafik ditampilkan:

- garis regresi populasi yang diperoleh dengan menggambar garis langsung dari dua parameter regresi yang diketahui;
- sebaran titik-titik berdasarkan data yang dibangkitkan;
- garis regresi penduga yang diperoleh berdasarkan estimasi parameter menggunakan fungsi lm();
- sabuk keyakinan prediksi garis regresi

Tampilan Jendela widget dan grafik yang dihasilkan adalah seperti gambar berikut.

2006





Gambar 5.6 Contoh Tampilan Grafik Dinamik. Tampilan memuat Jendela Widget sebagai pengatur parameter dan Jendela Grafik tempat tampilan grafik.

5.6 Grafik Dinamik dengan tkrplot

Kalau pada desai sebelumnya grafik ditampilkan pada jendela grafik seperti biasanya (untuk grafik pada umumnya), dengan tkrplot grafik danjendela widget berada pada satu bingkai. Dalam hal ini grafik diperlakukan sebagai image.



Langkah-langkah yang perlu dilakukan dalam memanfaakan tkrplot adalah sebagai berikut ini (dimodifikasi dari skrip demo tcltk).

1. **Menentukan fungsi yang akan digambar grafiknya, serta mengindentifikasi parameter yang diperlukan**. Misalnya yang akan digambar adalah grafik persepektif tiga dimensi. Dalam R bentuk fungsi perspektif ini adalah:

persp(x, y, z, theta = theta, phi = phi, expand =
 expand, r = r,d=d, ...))

2. Menentukan nilai awal dari masing-masing parameter, misalnya

persp(x,y,z, theta = 30,phi = 30,expand = 0.5, r = sqrt(3), d=1, ...

3. Mendefinisikan ulang fungsi grafik dalam format image widget

```
draw <- function() {
    par(bg = "white")
    try(persp(x, y, z, theta = theta, phi = phi,
    expand = expand, r = r,d=d, ...))
}</pre>
```

4. Menampilkan grafik sebagai bagian dari image widget

img<-tkrplot(base, draw)</pre>

5. **Membuat widget untuk mengatur paramete**r pada umumnya berupa slider

```
frame <- tkframe(base)
s.theta <- make.scale(frame, from=0, to=400,</pre>
```

2006

```
resolution=5, title="Kiri-Kanan", initial=theta,
           howvalue=TRUE, command=function(x) {
                     if (x != theta) {
                         theta <<- x
                         tkrreplot(img)
                     }
                 })
s.phi <- make.scale(frame, from=0, to=400,</pre>
          resolution=5,
                title="Atas-Bawah", initial=phi,
           showvalue=TRUE,
                command=function(x) {
                     if (x != phi) {
                         phi <<- x
                         tkrreplot(img)
                     }
                 })
s.expand <- make.scale(frame, from=0.05, to=2,</pre>
       resolution=0.05,title="Ekspansi", initial=expand,
       showvalue=TRUE, command=function(x) {
                    if (x != expand) {
                         expand <<- x
                         tkrreplot(img)
                     }
                 })
s.r <- make.scale(frame, from=0.05, to=15,</pre>
      resolution=0.05,
                title="Jarak Pandang", initial=r,
       showvalue=TRUE,
                command=function(x) {
                    if (x != r) {
                         r <<- x
                         tkrreplot(img)
                     }
                 })
s.d <- make.scale(frame, from=0.05, to=5,</pre>
      resolution=0.05,
                title="Efek", initial=d, showvalue=TRUE,
```

I Made Tirta

```
command=function(x) {
    if (x != d) {
        d <<- x
            tkrreplot(img)
        }
    })
tkpack(s.theta, s.phi, s.expand, s.r, s.d, side="left",
        anchor="n")
    tkpack(img,frame)
}</pre>
```

Tampilan grafik atau citra yang terjadi dapat dilihat pada Gambar 5.7.



Gambar 5.7. Tampilan Grafik Persepektif dengan tkrplot. Posisi gambar dapat diubah dengan menggeser-geser slider

5.7 Grafik Tiga Dimensi dengan RGL

R mempunyai paket khusus untuk menggambar objek tiga dimensi (3D). Paket ini disebut rgl. Selain mampu menggambar objek 3D R juga mampu menganimasikan objek yang dihasilkan secara mendasar melalui dua gerakan:

2006

- 1. memutar objek dengan cara menekan terus tombol mouse sebelah kiri sambil menggeser-geser mouse;
- 2. membesarkan atau mengecilkan ukuran objek dengan cara menekan terus tombol mouse sebelah kanan samber mengeser-geser mouse.

Berikut adalah contoh diagram pencar 3D yang menggambarkan kondisi suatu data hasil simulasi (Z=Ning, X=Nfis dan Y=Nmat, Subkelompok= Sekolah). Kemampuan ini dapat diakses melalui menu paket Rcmdr



Gambar 5.8. Contoh Gambar 3D dengan rgl. Pada tampilan ini dua posisi yang sedikit berbeda, (a) dan (b), diperoleh dengan mengerak-gerakkan mouse sambil menekan tombol mouse sebelah kiri.

I Made Tirta

Selain gambar 3D terkait dengan grafik statistika dan matematika umum, rgl dapat juga dimanfaatkan untuk membuat gambar 3D berupa objek 3D lain.



Gambar 5.9. Contoh Gambar Objek-objek 3D dengan rgl.

5.8 Desain Media Pembelajaran Statistika

5.8.1 Mendesain Sendiri Media Pembelajaran Statistika

Kemampuan grafik interaktif R dapat dimanfaatkan untuk mengkonstruksi media pembelajaran statistika yang berisi serangkaian ilustrasi grafik yang dapat memantapkan pemahaman mahasiswa terhadap konsep-konsep statistika. Di antara sekian konsep yang bermanfaat dilustrasikan dengan grafik interaktif adalah:

2006

- 1. Konsep interval keyakinan melalui ilustrasi peluang penutup (*coverage probability*)
- 2. Konsep konsistensi dan ketidakbiasan penduga dengan melakukan simulasi dengan ukuran sampel semakin besar
- 3. Konsep mean variansi dan korelasi dalam bivariat normal
- 4. Konsep penduga regresi sederhana maupun regresi dengan dua variabel penjelas
- 5. Konsep kedekatan distribusi t dan distribusi normal
- 6. Konsep teorema limit pusat (Lihat Tirta, 2005b)

Sebagai ilustrasi berikut adalah program dan tampilan untuk mengilustrasikan kedekatan antara distribusi T degan Z (normal baku). Langkah-langkah yang dapat ditempuh untuk membuat ilustrasi ini adalah seperti berikutini.

- 1. **Kajian teori**. Dari sisi teori kita tahu bahwa kedekatan antara T dan Z dipengaruhi oleh derajat kebebasan T. Oleh karena itu, salah satu variabel yang nilainya bisa digeser-geser adalah nilai derajat kebebesan T. Selain itu secara visual, kedekatan dua grafik sangat ditentukan oleh skala sumbu koordinatnya. Oleh kerane itu sumbu koordinat juga harus fleksibel diatur batas-batasnya.
- Menentukan fungsi yang akan digambar grafiknya, serta mengindentifikasi parameter yang diperlukan. Dalam hal ini yang akan digambar adalah grafik kepadatan dan kumulatif T dan Z. Dalam R nilai fungsi kepadatan diperoleh melalui ddistribusi dan nilai kumulatif melalui pdistribusi.

```
replot <- function(...) {
    if (is.null(y)) return() # too early...
    plot(x,y,type='l',col="red",main="Kepadatan Z dan
    T",ylim=c(yb,ya), xlab='X',ylab='P(x)',lty=2)
    lines(x,t,col="blue")
    plot(x,yp,type='l',col="red",main="Kumulatif Z
    dan T",ylim=c(yb,(ya+.5)),xlab='X',ylab='P(x)',
        lty=2)
    lines(x,tp,col="blue")</pre>
```

I Made Tirta

- Menentukan nilai awal dari masing-masing parameter, misalnya nilai awal batas sumbu koordinat (misalnya X=(-6,6), Y=(0,0.45), derajat kebebasan T (misalnya 2)
- Membuat widget untuk mengatur parameter pada umumnya berupa slider. Lay-out Desain yang dapat dipilih adalah berupa frame seperti berikut.

Derajat Kebebasan		
Batas Absis X (Frame kiri)	Batas Ordinat (Framekanan)	
X0 (frame1)	Y0 (frame3)	
X1 (frame2)	Y1 (frame4)	
Keluar/tutup		

Program untuk desain di atas adalah

```
base <- tktoplevel()</pre>
    tkwm.title(base, "Distribusi Z dan T")
    spec.frm <- tkframe(base,borderwidth=2)</pre>
    left.frm <- tkframe(spec.frm)</pre>
    right.frm <- tkframe(spec.frm)</pre>
    ## Two left frames:
    frame1 <-tkframe(base, relief="groove", borderwidth=2)</pre>
    tkpack(tklabel(frame1, text="Derajat Bebas T"))
                                  command=regen,
        tkpack(tkscale(frame1,
                                                       from=1,
to=600.00,
                    showvalue=T, variable=db,
                    resolution=1, orient="horiz"))
  frame2 <-tkframe(left.frm, relief="groove", borderwidth=2)</pre>
  tkpack(tklabel(frame2, text="Batas X"))
  tkpack(tkscale(frame2, command=regen, from=-7, to=0,
                    showvalue=T, variable=x1,
                    resolution=0.1, orient="horiz"))
  tkpack(tkscale(frame2, command=regen, from=0, to=7,
```

2006

```
showvalue=T, variable=xu,
                   resolution=0.5, orient="horiz"))
                                            relief="groove",
  frame3
                <-tkframe(right.frm,
borderwidth=2)
  tkpack(tklabel(frame3, text="Batas Y"))
  tkpack(tkscale(frame3, command=regen, from=-0.1, to=0.4,
                   showvalue=T, variable=yl,
                   resolution=0.01, orient="horiz"))
  tkpack(tkscale(frame3, command=regen, from=0.2, to=0.95,
                   showvalue=T, variable=yu,
                   resolution=0.01, orient="horiz"))
  tkpack(frame1, fill="x")
  tkpack(frame2,frame3, fill="x")
  tkpack(left.frm, right.frm,side="left", anchor="n")
  #`Bottom frame' (on base):
  q.but <- tkbutton(base,text="Selesai",</pre>
                      command=function()tkdestroy(base))
 tkpack(spec.frm, q.but)
```

I Made Tirta





5.8.2 Memanfaatkan Paket TeachingDemos

2006

Mulai R versi 2.5 juga sudah ada paket yang secara khusus dikembangkan untuk pembelajaran statistika yang dikemas dalam paket TeachingDemos. Paket ini dikembangkan oleh Greg Snow. Beberapa ilustrasi pembelajaran
Probabilitas dan Statistika yang telah dibuat di antaranya adalah (Lihat Gambar ... dan Gambar ...):

- 1. ilustrasi toss uang logam dan dadu dengan ilustrasi 3D memanfaatkan paket rgl;
- 2. ilustrasi Teorema limit pusat;
- 3. ilustrasi konsep likelihood maksimum;
- 4. lustrasi regresi pohon



Gambar 5.11. Ilustrasi Tos Uang Logam. Dengan rgl dapat diilustrasikan animasi putaran atau gerakan uang logam



Gambar 5.12. Ilustrasi Tos Dadu. Dengan rgl dapat diilustrasikan animasi putaran atau gerakan dadu

I Made Tirta

Demo-demo untuk pembelajaran juga dapat di-plug-in ke dalam menu Rcommander seperti dilihat pada Gambar Gambar 1.17.

5.9 Dari Menu ke CLI

Pada bagian ini akan dibahas cara menerjemahkan pilihan pada menu menjadi perintah skrip sebagaimana kita menggunakan CLI. Pembahasan ini merupakan bagian yang sangat penting dalam membuat menu GUI dari paket atau perintah yang biasanya dipanggil melalui perintah skrip.

substitute(perintah CLI lengkap)

```
x <- parse(text=tclvalue(xvar))[[1]]
y <- parse(text=tclvalue(yvar))[[1]]
substitute(print(summary(lm(y~x))))
```

Skrip lengkap pemrograman ini dapat dilihat pada lampiran. Tampilan widget dapat dilihat pada Gambar 5.13 dan hasilnya seperti berikut ini.

```
Call:
lm(formula = Orange$circumference ~ Orange$age)
Residuals:
               10
                                   30
                    Median
    Min
                                            Max
-46.31030 -14.94610 -0.07649 19.69727 45.11146
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 17.399650 8.622660 2.018 0.0518
Orange$age 0.106770 0.008277 12.900 1.93e-14 ***
___
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
Residual standard error: 23.74 on 33 degrees of freedom
Multiple R-Squared: 0.8345, Adjusted R-squared: 0.8295
F-statistic: 166.4 on 1 and 33 DF, p-value: 1.931e-14
```

2006

🛪 Regre	si	
	Analisis Regresi	
x variable	Orange\$age	
y variable	Orange\$circum	ference
Kirim	Rese	t

Gambar 5.13. Tampilan Jendela Data untuk Analisis Regresi

5.10 Bacaan Lebih Lanjut

R merupakan program *open source* yang di Indonesia belum banyak dimanfaatkan. Oleh karena itu mungkin agak kesulitan mencari ahli pemrograman maupun referensi berbahasa Indonesia. Berikut adalah tips untuk mengatasi kendala tersebut.

- 1. Jelajahi situs internet yang memuat R-GUI. Salah satu alamat ini adalah <u>http://www.sciviews.org/ R-GUI/tcltk/index.html</u> dan lihat contoh-contoh yang ada.
- 2. Praktekkan skrip contoh yang ada, selanjutnya lakukan modifikasi di sana sini sampai anda memahami struktur pemrogramannya.
- 3. Untuk kepentingan dikemudian hari, skrip yang anda buat diberi catatan atau komentar (dengan menambahkan tanda # didepannya)

5.11 Latihan

- 1. Buatlah satu Widget yang lengkap memuat komponen berikut, dengan memanfaatkan kemampuan paket tcltk, dan tcltk2
 - a. Menu lepas dan tidak lepas

I Made Tirta

- b. Menu dengan satu atau dua tingkat
- c. Menu yang terhubung dengan salah satu paket R
- d. Jendela untuk menulis skrip
- e. Pilihan Keluar baik melalui menu maupun tombol
- 2. Modifikasi skrip demo tcltk sehingga mampu menghasilkan tampilan widget dan grafik berikut







Laboratorium Statistika-Jurusan Matematika FMIPA UNEJ

6 MENGEKSPLORASI DAN MEMODIFIKASI PAKET

Setelah kita belajar mendesain sendiri widget dengan berbagai variasi, sebagai bagian dari tahapan melatih diri, pada bab ini kita akan mengeksplorasi kemampuan paket-paket yang telah dilengkapi dengan fasilitas GUI serta memodifikasi ata mengembangkan untuk kepentingan sendiri. Disarankan agar sebelum memulai memodifikasi paket-paket yang telah ada, pembaca memahami dengan baik dasar-dasar pemrograman menggunakan R-Tcl/Tk seperti diuraikan sebelumnya. Scara khusus diharapkan agar setelah membaca uraian pada bab ini pembaca dapat mengeksplorasi dan memanfaatkan kemampuan GUI dari paket PBSmodelling, paket TeachingDemos, dan paket Rcmdr.

6.1 Mengeksplorasi Paket

Sebagaimana telah disebutkan sebelumnya, bahwa ada beberapa paket R yangtelah dilengkapi dengan kemampuan GUI yang pada dasarnya juga dikembangkan dari Paket Tcl/Tk. Sebagaimana prinsip *open source,* bahwa seseorang tidak harus memulai sesuatu dari awal, namun dia dapat memanfaatkan paket yang telah ada untuk dikembangkan atau dimodifikasi sesuai keperluan. Ada tiga paket yang telah memanfaatkan aplikasi Tcl/Tk yaitu:

- 1. Paket PBSmodelling. Paket ini sesungguhnya dibuat untuk menganalisis data hayati di pasifik (*Pacifik Biological Station*), namun fungsi-fungsi untuk membuat tampilan GUI (biasa disebut Widget) dibuat sedemikain rupa sehingga mudah dimanfaatkan untuk keperfluan lain.
- 2. TeachingDemos. Paket ini secara khusus dikembangkan untuk membantu penanaman konsep peluang dan statistika. Paket ini banyak memuat ilustrasi simulasi konsep-konsep statistika.

I Made Tirta

3. Rcommander. Paket Rcommander adalah paket analisis data yang memanfaatkan TclTk untuk membangun menu. Paket ini sangat bermanfaat bagi pengguna statistika dasar yang biasa menggunakanmenu. Selain dapat dijadikan model pengembangan analisis data, paket ini juga telah tersedia dalam menu bahasa Indonesia.

6.2 Membuat Widget dengan Paket PBSmodelling.

Melalui paket PBSmodelling pembuatan widget menjadi lebih sederhana. Beberapa fungsi yang telah disediakan untuk membuat widget. Secara umum program untuk membuat widget dibedakan menjadi dua komponen, yaitu:

1. **Fungsi-fungsi R**. Bagian ini berisi sekumpulan fungsi-fungsi yang diperlukan atau akan dimanfaatkan dalam suatu widget. Perintah membangun jendela (widget) dipanggil melalui file ini, tetapi deskripsi widgetnya dipisahkan pada bagian deskripsi.

```
myPlot <- function() {
getWinVal(scope="L");
definisi dan tugas lain yang diperlukan
};</pre>
```

 Deskripsi jendela. Bagian ini berisi perintah mendasar untuk membangun widget. Saat ini, melaui bagian ini, pengguna dapat mendefinisikan sekitar 18 komponen widgets (mulai dari jendela utama, menu, tombol radio, penggeser dan sebagainya, lihat Schnute *et al.* 2006). Untuk wigget sederhana, bagian (1) dan (2) dapat dituliskan pada satu file dengan memberi pilihan (astext=TRUE)

```
winStr=c(
"window title=\"Judul Jendela\"",
"button function=NamaFungsi text=\"Judul Teks \"");
```

3. File deskripsi Jendela (*windows desrcription*). Sama dengan (2) di atas hanya untuk widget yang lebih kompleks bagian ini biasanya dan

```
2006 Laboratorium Statistika-Jurusan Matematika FMIPA UNEJ
```

lebih mudah dibuat terpisah menjadi satu file tersendiri menggunakan ekstensi txt dan pada pembuatan widget diberi opsi (astext=astext) atau tanpa opsi karena nilai *default*-nya sudah demikian

4. Perintah untuk membuat jendelal adalah createWin(), yang ditulis pada gabian akhir dari file deskripsi fungsi-R. Untuk deskripsi yang bergabung dalam satu file padabagian akhir file ini biasanya berisi perintah berikut.

require(PBSmodelling); createWin(winStr,astext=TRUE)

Sedangkan untuk deskripsi jendela yang terpisah dalam file tersendiri, dibangun dengan perintah berikut.

```
createWin(system.file("NamaFile.txt",package="PBSmodelling"))
```

Berikut adalah skrip lengkap beserta widget dan grafik yang dihasilkan masing masing disajikan pada Gambar 6.1dan Gambar 6.2.

```
#diskripsi jendela
winStr=c(
"window title=\"Contoh Widget Mini\"",
"button function=myPlot text=\"Plot Diagram
Pencar\"");
# function yang diperlukan
myPlot <- function() {
getWinVal(scope="L");
x <- seq(10,50,1);
y<-3+2*x+rnorm(length(x),0,5);
plot(x,y,type="p",main="Diagram Pencar",xlab="X",
ylab="Y",col='blue',pch='*');
yl<-3+2*x;
lines(x,y1,col='red');
addArrows(0.4,.8,0.5,.55,col='green')</pre>
```

I Made Tirta

```
addLabel(0.4,.85,"Tambah Teks")
};
# perintah membangun jendela (tergabung)
require(PBSmodelling); createWin(winStr,astext=TRUE)
```



Gambar 6.1 Widget Sederhana dengan PBSModelling

2006



Gambar 6.2 Tampilan Grafik Dihasilkan dari Widget Sederhana

Contoh berikut aberisi beberapa komponen widget yang dihasilkan dengan deskripsi terpisah. Karena fokus yang mau diilustrasikan adalah komponen widgetnya, fungsi perintah dalam file ini dibuat dummy (my.fun). Pembaca dapat membuat fungsi yang sesuai keperluan. Karena pembuatan widget dengan PBSmodelling memerlukan paket PBSmodelling, maka lebih mudah jika kumpulan widget ini dibuat pada

I Made Tirta

direktori library PBSmodelling. Penulis menggunakan direktori New, sehingga pada akhir file R tertulis

```
require(PBSmodelling)
my.fun<-function(){
    cat("\nTes\n")
}
createWin(system.file("New/ContohWin1.txt",package="PBSmodel
ling"))</pre>
```

Sedangkan pada file ContohWin1.txt dapat ditulis

```
window title="Contoh Jendela"
#membuat menu dengan 1 submenu dan 4 submenu
menu nitems=1 label="Menu1"
  menuitem label="Item menu1" func=my.fun
menu nitems=4 label="Menu2"
  menuitem label="item menu21" func=my.fun
  menuitem label="item menu22" func=my.fun
  menuitem label="item menu23" func=my.fun
  menuitem label="item menu24" func=my.funS
#mendefinisikan grid 2x3
Grid 2 3 sidetitle="MATRIKS GRID 2x3" sidefont="bold 10"
Label A11
Label A12
Label A13
Label B21
Label B22
Label B23
#Mendefinisikan tombol dengan layout 1x6
grid 1 6 sticky=W
 label text=Tombol: font=bold
 button text=Tombol1 func=my.fun
 button text=Tombol2 func=my.fun
 button text=Tombol3 func=my.fun
```

Laboratorium Statistika-Jurusan Matematika FMIPA UNEJ

```
button text="Tombol 4" func= my.fun
 button text=Tombol5 func= my.fun
#Membuat frame dengan layout 2x2
label text="Frame" font=bold
grid 2 2 relief=groove toptitle=Kolom sidetitle=Baris \
   topfont="Helvetica 12 bold" sidefont="Helvetica 12 bold"
      label text="Sel 1" font="times 8 italic"
      label text="Sel 2" font="times 10 italic"
      label text="Sel 3" font="times 12 italic"
      label text="Sel 4" font="times 14 italic"
# Membuat slide
label text="Slide" font=bold
slide name=junk from=1 to=1000 value=225 showvalue=T
slideplus name=junk2 from=0 to=1 by=0.01 value=0.75 padx=10
#membuat tombol radio dengan layout horizontal (1x5)
grid 1 5 sticky=W sidetitle="Tombol Radio" sidefont="bold
10"
     radio name=dset text="radio 1" padx=5 mode=character
            value=sim func= my.fun
     radio name=dset text="radio 2" padx=5 mode=character
            value=cars func= my.fun
     radio name=dset text="radio 3" padx=5 mode=character
            value=trees func= my.fun
     radio name=dset text="radio 4" padx=5 mode=character
            value=swiss func= my.fun
     radio name=dset text="radio 5" padx=5 mode=character
            value=attitude func= my.fun
#membuat tombol cek dengan layout 1x2
grid 1 2 sidetitle="Tombol Cek" sidefont="bold 10"
check name=yep checked=T text="Cek di sini" padx=10 pady=10
check name=nope checked=F text="atau tidak" padx=10 pady=10
#membuat vektor dengan nilai dan ceklis
vector length=4 names="a b g d" labels="alpha beta gamma
    delta" \
  values="100 0.05 1 5" font="times italic" width=6
vector length=5 mode=logical names=chosen labels=pilih \
```

I Made Tirta

```
values="F T F T T" vertical=F
#membuat matriks
matrix nrow=2 ncol=3 rowlabels="'Row A' 'Row B'" \
    collabels="'Col 1' 'Col 2' 'Col 3'" \
values="10 20 30 100 200 300" names="a b c d e f" \
    font="times 10 italic"
```

Label end

6.3 Menerjemahkan dan Memodifikasi Menu Rcmdr

1.9.1 Menerjemahkan menu dan dialog RCommander

Rcmdr atau RCommander adalah salah satu paket R yang berbasis R-GUI. RCommander yang dikembangkan oleh Prof J. Fox aslinya adalah berbahasa Inggris. Belakangan dengan bekerja sama dengan beberpa pengguna RCommader diberbagai negara, menu dan dialog RCommander tersedia dalam berbagai bahasa, termasuk bahasa Indonesia yang merupakan kontribusi penulis (Lihat Gambar 6.4). Terjemahan menu dilakukan melalui file yang berekstensi ".po". Untuk terjemahan menu bahasa Indonesia biasanya terletak pada subdirektori "...\Rcmdr\po\id". Struktur terjemaannya adalah seperti berikut:

msgid "Not enough memory" # teks asli bahasa inggris
msgstr "Memori tidak cukup" # teks terjemahan

2006

7 Contoh Jendela	
Menu1 Menu2	
Item menu1 MATRIKS GRID 2x3 A11 A12 A13 B21 B22 B23	
Tombol: Tombol2 Tombol3 Tombol 4 Tombol5	
Frame	
Kolom	
Sel 1 Sel 2	
Sel 3 Sel 4	
Slide	
225	
Min-> 0 0.75 1 <-Max	
Tombol Radio (© radio 1 C radio 2 C radio 3 C radio 4 (C radio 5
Tombol Cek 🔽 Cek disini 🗖 atau tidak	
alpha beta gamma delta	
1100 JU.US J1 J5	
pilih	
Col 1 Col 2 Col 3	
Row A 100 0.05 30	
200 I300	
end	



1.9.2 Memodifikasi dan Menambahkan Menu RCommander

Selain terjemahan melalui fasilitas file terjemahan, kita dapat juga melakukan modifikasi atau pengembangan pada menu RCommander. Gambar berikut adalah tampilan RCommander dalam Bahasa Inggris,

I Made Tirta

dalam Bahasa Indonesia dengan menu baku dan dalam bahasa Indonesia dengan menu tambahan.



Gambar 6.4. Kontributor Terjemahan RCmdr ke Berbagai Bahasa



Gambar 6.5. Tampilan RCommander Baku dalam Bahasa Indonesia

Untuk melakukan modifikasi pada menu kita harus memahami dua file utama yang dimiliki Rcmdr yang terletak pada direktori pustaka Rcmdr yaitu:

 Rcmdr-menus.txt pada direktori Rcmdr\etc. File ini berisi struktur menu yang dimiliki Rcmdr. Adapun struktur menu dan item menu adalah

2006

2. **Rcmdr** pada direktori **Rcmdr****R**. File ini berisi fungsi-fungsi yang terkait dengan menu Rcmdr.

Langkah-langkah untuk menambahkan menu atau item menu pada Rcommander adalah sebagai berikut:

1. Pada file Rcmdr-menus.txt tambahkan menu atau item menu. Tambahan menu utama, merupakan item jenis cascade yang ditempelkan pada topMenu sehingga formatnya adalah

menu	NamaMenu	topMenu			
item	topMenu	cascade	"LabelMenu"	NamaMenu	

2. Selanjutnya menu utama tadi diisi item-item yang sesuai dengan keperluan, item ini termasuk jenis command dan memiliki format

3. Yang terakhir adalah mendefinisikan Fungsi perintah pada file Rcmdr. Fungsi yang dipanggil dan yang didefinisikan harus sama persis (ingat bahwa R termasuk program yang *case sensitive*, sehingga penggunaan huruf kecil atau kapital harus persis sama.

FungsiPerintah<-function(){</pre>

..... J

LabelMenu adalah nama yang akan muncul pada jendela Widget., sedangkan NamaMenu adalah nama yang dipergunakan untuk memanggil menu yang baru dibuat. Misalkan kita ingin menambahkan menu "Demo" dengan salah satu itemnya adalah demo bootstrap yang berada pada paket boot, maka pada file Rcmdr-menus.txt dapat dituliskan

item demoMenu command "Demo Bootstrap" bootDemo

Sedangkan pada file Rcmdr dapat didefinisikan perintah bootdemo sebegi berikut.

I Made Tirta

```
bootDemo <- function() {
    require(boot)
    example(plot.boot)
    }</pre>
```

Dengan sedikit modifikasi, pembaca dapat juga menambahkan submenu pada menu yang ditambahkan. Tampilan hasil modifikasi menu pada Rcommader dapat dilihat pada Gambar 6.6. Pada gambar tersebut menu "Demo*" memiliki submenu Äneka Rupa" sedangkan item bootstrap masuk dalam submenu "Aneka Rupa".



Gambar 6.6. Tampilan RCommander yang Telah Dimodifikasi

1.9.3 Menambahkan Menu Analisis

Apabila kita telah dapat memahami struktur menu GUI Rcommander, kita mungkin menambahkan menu analisis yang CLI nya sudah ada, tetapi belum dimasukkan dalam GUI, misalnya analisis GEE. Untuk itu dapat dilakukan langkah-langkah berikut.

1. Pilih menu yang paling dekat, misalnya untuk GEE (karena ada komponen distribusi, link) maka menu yang paling dekat adalah menu dialog GLM.

2006

- 2. Salin fungsi yang paling dekat tadi dan beri nama baru sesuai keperluan (misalnya dari GeneralizedLinearModel menjadi geeModel).
- 3. Lakukan modifikasi yang diperlukan serta tambahan item dalam dialog. Misalnya untuk GEE selain distribusi dan link juga diperlukan klaster dan struktur korelasi.
- 4. Lihat juga bagian-baian widget yang terkait. Misalnya karena klaster pada dasarnya adalah faktor, maka kita dapat melihat, menyalin dan memodifikasi tampilan faktor pada anova
- 5. Pada bagian perintah lakukan modifikasi sesuai perintah CLI dari analisis yang kita lakukan. Misalnya perbedaan antara perintah CLI antara GLM dan GEE adalah seperti berikut

Dengan demikian pada bagian akhir perintah glm dapat diganti dengan gee seperti pada contoh berikut (perhatikan beda keduanya).

command <- paste("glm(", formula, ", family=", family, "(", link, "), data=", ActiveDataSet(), subset, ")", sep="") command <- paste("gee(", formula, ", family=", family, "(", link,"), id=", id, ", corstr=\"",cor,"\", data=", ActiveDataSet(), subset, ")", sep="")

Tanda \" diperlukan karena format struktur korelasi pada gee menggunakan format karakter yang ditulis diantara tanta petik ganda.

I Made Tirta

i Pendugaan P	ersam	aan T	ergen	eralis	ir (GE	E))			
Masukkan nama u	intuk m	odel: G	EE.1						
Peubah (klik dobe	luntuk	Iormula)						
age circunterence Tree [faktor]		8							
Formula Model:	+	*		1	%in%	-	*	(
~									
41 XI	301								1
Ekspresi bagian									
<semua kasus="" yar<br="">Si Famili (klik-dobel u</semua>	ng valid. Intuk me	emilih)	Fun	gsi hub	ungan (link)			
gaussian	_		logit	ji F					
poisson			cloc	loa					
Gamma				-					
Klaster (pilih satu)			Stru	ktur Ka	relasi (p	ilin satu	J)		
Tree		<u>s</u>	inde AR- excl	pender 1 hanges	nce Ible				
				a state of the state of the			0.000		

Gambar 6.7. Tampilan Dialog GEE. Dialog ini memiliki dua item tambahan diabandingkan dengan dialog GLM, yaitu klaster dan struktur korelasi.

6.4 Mengemas Paket

Program-program yang telah dibuat dapat dikemas menjadi suatu paket yang dapat didistribustikan dan diinstal sehingga dapat dimanfaatkan orang lain. Untuk dapat didistribusikan, program harus dilengkapi dengan:

- 1. definisi fungsi-fungsi yang dibuat
- 2. petunjuk pemakaian dan dokumentasi baik yang bersifat umum maupun yang bersifat kusus dari masing-masing subprogram;
- 3. cara mengeksekusi program;
- 4. contoh-contoh pemakaian dari masing-masing subprogram.

2006

6.4.1 Menyiapkan Fungsi-fungsi Terkait

Fungsi-fungsi yang didefinisikan dapat dikumpulkan dalam satu file atau dalam beberapa file terpisah tergantung pada banyaknya dan kompleksnya fungsi. File dapat diberi nama sesuai fungsi-fungsi yang dimuatnya. Pengemasan seperti ini akan lebih membantu orang lain yangi ngin memanfaatkan atau mengembangkan lebih jauh fungsi-fungsi yang kita buat.

6.4.2 Menyiapkan Dokumen Bantuan

Dokumen bantuan R harus mengikuti format berikut.

- 1. **Header** yang berisi judul atau nama paket (misalnya **glm**).
- 2. Judul Dokumen (misalnya *Fitting Generalied Linear Models*).
- 3. **Description**, yaitu menguraikan deskripsi (menjelaskan fungsi, penggunaan dan lain-lain) paket.
- 4. **Usage,** menjelaskan cara penggunaan (cara memanggil paket) paket. Misalnya glm(....,...)
- 5. **Arguments**, menjelaskan cara mengisi parameter atau komponen dari fungsi yang diuraikan pada bagian *usage* dia atas. Misalnya pada glm(), ada komponen yang wajib diisi yaitu formula, family dan data.
- 6. **Details**, menguraikan lebih rinci cara penggunaan atau pemanggilan fungsi terkait dengan *usage* dan *arguments* yang telah diuraikan.
- 7. **Values**, menguraikan keluaran yang dihasilkan oleh program atau paket yang dijalankan. Misalnya untuk glm(), keluaran yang dihasilkan berupa koefisien regresi berserta komponen lainnya berupa sisa, nilai pengepasan dan sebagainya.
- 8. Authors, memuat nama-nama pengebang program atau paket.
- 9. **Referensi,** memuat daftar oustaka terkait dengan pakety yang dikembangkan.
- 10. **See Also,** menguraikan paket atau sub paket (fungsi-fungsi R) yang terkait.
- 11. **Examples,** berisi contoh lengkap penggunaan paket. Bagian ini berisi skrip contoh yang minimal dapat di*copy-paste* ke Console R.

6.4.3 Mengemas Dokumen Bantuan

Dokumentasi R dihasilkan melalui pengolah kata LaTeX. Oleh karena itu, dokumen dengan format yang dijelaskan sebelumnya harus ditulis dalam teks ascii dengan struktur berikut. Bagi penulis yang telah biasa bekerja dengan LaTeX akan biasa mlihat struktur dokumen berikut.

\name{nama}

Nama yang dipakai harus unik tidak boleh tumpang tindih dengan nama paket yang lain.

\alias{topik}

Bagian ini berguna membantu R membuat indeks, selain itu melalui topik ini paket kita dapat dilacak, karena itu berikan lebih dari satu topik tertait. Misalnya untuk pembangkitan data acak normal, berikut adalah topik, topik yang semuanya dapat dimasukkan.

```
\name{Normal}
\alias{Normal}
\alias{dnorm}
\alias{pnorm}
\alias{qnorm}
\alias{rnorm}
```

\title{Judul}

Judul harus menggunakan *title case* dan tidak diakhiri titik. Hindarkan penggunaan simbol-simbol matematika atau lainnya (misalnya <,> dan lain-lainnya).

```
\description{deskripsi singkat paket yang dibuat}
\usage{fungsi(argumen1, argumen2, ...)
}
```

Berisi cara memanggil fungsi dengan tepat. \usage{glm(formula,family,data,...)

```
\arguments{ deskripsi argumen fungsi seperti diuraikan
sebelummnya}
```

Deskripsi argumen disajikan dalam format
\item{formula}{deskripsi formula }

```
\item{ioimula}{deskripsi ioimula
\item{family}{deskripsi family)
```

Laboratorium Statistika-Jurusan Matematika FMIPA UNEJ

```
detail tentang
\details{uraian yang agak
                                                  paket,
argumen yang diperlukan.}
\value{ Berisi uraian
                            tentang
                                       keluarasan
                                                    yang
dihasilkan oleh paket.}
\references{Daftar
                     pustaka-pustaka
                                       perkait
                                                  dengan
paket yang dikembangkan}
\note{Catatan khusus jika diperlukan}
\author{Nama-nama pengembang paket}
\seealso{Paket atau fungsi terkait}
      Gunakan \code{\link{...}} untuk menghasilkan dokumen yang
      secara otomatis link dengan bagian yang ditunjuk.
\examples{Contoh penggunaan paket}
\keyword{kata kunci}
```

Hal yang perlu diperhatikan dalam membuat dokumen adalah ukuran dokumen jangan terlalu besar. Jika terlalu besar, sebaiknya fungsi dan dokumentasinya dipecah menjadi beberapa fungsi dan dokumen yang lebih kecil sehingga lebih mudah difahami.

6.4.4 Menyiapkan Program Pendukung

Untuk lebih memperlancar pengemasan paket, program-program komputer berikut wajib tersedia dalam komputer tem[at kita mengemas paket, yaitu:

- 1. Program R. Usahakan instal programm R versi terbaru sehingga pendistribusian paket yang kita buat tidak mengalami kendala jika dijalankan orang lain pada program R dengan versi terbaru. R versi terbaru dapat diunduh dari di http://www.cran.r-project . org dengan memilih situs *mirror* terdekat.
- 2. *RTools* versi windows. Paket program ini dapat diunduh dari <u>http://www.murdoch-sutherland.com/Rtools/</u>
- 3. Program Perl yang dapat diunduh dari <u>http://www.activestate.com/Products/ActivePerl/Download.html</u>

I Made Tirta

- 4. MinGW, program minimum versi GNU untuk mengkompilasi C untuk Windows. Program ini dapat diunduh dar situs <u>http://www.mingw.org/</u>
- 5. MikTex dan WinEdt untuk mengkompilasi dokumen bantuan. Kedua program ini masing-masing dapat diunduh dari <u>http://miketex.org/</u> dan http://www.winedt.com/
- HTML help Workshop untuk mengkompilasi bantuan dokumen format html. Program ini dapat diunduh dari <u>http://www.microsoft.com/office/ork/xp/appndx/appa06.htm</u> (Informasi lebih detail dapat dilihat pada Schnute *et. al,* 2006).

6.4.5 Langkah Mengemas Paket

Sebelum memulai mengemas paket terlebih dahulu siapkan direktori tempat kita menyimpan paket-paket yang kita kemas, misalnya D:\PaketBaru. Selanjutnya buka paket PBStry_x.xx.tar.gz dan salin/copy file-file berikut (check.bat, build.bat, definePaths.bat, unpack.bat, makePDF.bat) ke direktori ini. Selanjutnya untuk mengemas paket "PaketA" lakukan langkah berikut:

 Periksa paths program-program pengemas pada file definePaths.bat dan sesuaikan dengan letak path yang sesungguhnya, misalnya D:\util dengan C:\Program Files atau yang lain sesuai tempat program tersebut diinstal

```
set R_PATH=d:\util\R-2.5.0\bin
set TOOLS_PATH=d:\Util\Rtools\bin
set PERL_PATH=d:\Util\Perl\bin
set MINGW_PATH=d:\Util\MinGW\bin
set TEX_PATH=c:\Program Files\MikTeX 2.5\miktex\bin
set HTMLHELP_PATH=d:\Util\HTMLHW
```

2. Buat subdirektori D:\PaketBaru\A dan beberapa subditerktori bertikut:

```
D:\PaketBaru\PaketA\man
D:\PaketBaru\PaketA\R
D:\PaketBaru\PaketA\inst
D:\PaketBaru\PaketA\data
Letakkan atau salin file zzz.R dari ...\PBStry\R ke ...\PaketA\R
```

2006

3. Buat file ascii D:\PaketBaru\PaketA\DESCRIPTION (tanpa ekstensi) yangberisi informasi berikut

```
Package: Nama_Paket (Paket A)
Version: x.x.x (format 3 digit)
Date: Tanggal penulisan paket
Title: Judul Paket
Author: Nama<nama@email>
Maintainer: Nama<nama @email>
Depends: R (>= 2.4.0)
Description:Deskripsisingkat.
License: GPL version 2 or newer
Packaged: tanggal dikemas; Pengemas
```

Nama paket yang ditulis dalam file DESCRIPTION harus sama persis dengan nama direktori tempat menyimpan file-file yang akan dikemas.

- 4. Letakkan file-file fungsr R pada ...\PaketA\R\ dan file-file dokumentasi bantuan pada ...\PaketA\man\
- 5. Selajnutnya secara berurutan dipanggil checkpath nama_paket build nama_paket makePDF nama_paket
- 6. Informasi lebih detail dapat dilihat pada Schnute *et. al,* 2006.

6.5 Bacaan Lebih Lanjut

Saat ini sudah ada cukup banyak paket-paket R yangmemanfaatkan dan mengembangkan RGI. Berikut adalah tips untuk memanfaatkan paket terkait.

- 7. Lacak paket-paket yang memanfaatkan GUI (yang telah dibahas misalnya Rcmdr, PBSModeling, Gasper)
- 8. Pelajari *source* paket-paket R yang memuat GUI tersebut (misalnya Rcmdr, tcltk, tcltk2, tkrplot, PBSmodelling, TeachingDemos).
- 9. Coba contoh-contoh atau demo yang ada pada paket di atas dan berusahalah memahami struktur pemrogramannya dengan memodifikasi beberapa parameter sampai anda memahaminya.

I Made Tirta

10. Bila anda memodifikasi skrip program untuk dipublikasikan lagi, jangan lupa **dua hal penting** yaitu (i) tetap mencantumkan atau merujuk penulis awal (aslinya) dan (ii) menunjuk bagian yang dimodifikasi.

6.6 Latihan

- 1. Buatlah satu Widget yang lengkap memuat komponen berikut, dengan memanfaatkan peket lain (misalnya PBSmodelling):
 - a. Menu dengan satu atau dua tingkat
 - b. Menu yang terhubung dengan salah satu paket R
 - c. Jendela untuk menulis skrip
 - d. Memuat Tombol radio dan penggeser
 - e. Pilihan Keluar baikmelalui menu maupun tombol
- 2. Modifikasi menu Rcommander dengan menambahkan menu Demo beberapa paket analisis yang menurut anda menarik.
- 3. Eksplorasi paket TeachingDemos lalu modifikasi untuk kepentingan ilustrasi pembelajaran anda di kelas
- 4. Untuk latihan mengemas paket lakukan:
 - a. Buat definisi beberapa fungsi sederhana,
 - b. buat dokumentasi terkait,
 - c. selanjutnya lakukan pengemasan paket.

2006

DAFTAR PUSTAKA

- Dalgaard, P. 2001. A Primer on the R-Tcl/Tk Package. *R-News*. 2001-3. 27-31.
- Dalgaard, P. 2002. Changes to the R-Tcl/Tk package. *R-News*. 2002-3. 25-27
- Fox, J (dengan kontribusi Michael Ash, Philippe Grosjean, Martin Maechler, Dan Putler and and Peter Wolf) . 2005. *Rcmdr: R Commander. R package version 1.1-7.* <u>http://www.r-project.org</u>,

http://socserv.socsci.mcmaster.ca/jfox/Misc/Rcmdr/

- Grosjean, P. 2005. *tcltk2: SciViews GUI API More tcltk*. R package version 0.9-5. http://www.sciviews.org/SciViews-R
- Grosjean, P. 2006. *R tcltk coding examples*. <u>http://www.sciviews.org/_R-GUI/tcltk/ index.html</u> 20 Agustus 2006].
- R Development Core Team (RDCT). 2005. R: *A language and environment* for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <u>http://www.R-project.org</u>.
- R Development Core Team (RDCT). 2006. *FAQ for Windows Port.* Manual elektronik R 2.2
- Schnute, J.T. Couture-Beil, A. & Haigh R. 2006. *PBS Modelling 1: User's Guide*. Cannadian Technical Report on Fisheries and Aquatic Science. 2674
- Schnute, J.T. Couture-Beil, A. & Haigh R. 2007. *PBS Modelling 1:* Developer's Guide.
- Tirta. I.M. 2005a. *Panduan Program Statistika R* UPT Penerbit Universitas Jember (ISBN :979 -8176-37-5)

I Made Tirta

- Tirta. I.M. 2005b Potensi dan Prospek Pemanfaatan OSS-R dalam Analisis Data dan Pengajaran Statistika, *Pancaran Pendidikan* No. 61: 195-208 (terakreditasi).
- Zhang.J. 2007. Examples of Somes of the Widgets. Bioconductor,.

2006

LAMPIRAN

glm(stats)

R Documentation

Fitting Generalized Linear Models

Description

glm is used to fit generalized linear models, specified by giving a symbolic description of the linear predictor and a description of the error distribution.

Usage

```
glm(formula, family = gaussian, data, weights, subset,
    na.action, start = NULL, etastart, mustart,
    offset, control = glm.control(...), model = TRUE,
    method = "glm.fit", x = FALSE, y = TRUE, contrasts = NULL, ...)
glm.fit(x, y, weights = rep(1, nobs),
    start = NULL, etastart = NULL, mustart = NULL,
    offset = rep(0, nobs), family = gaussian(),
    control = glm.control(), intercept = TRUE)
## S3 method for class 'glm':
    weights(object, type = c("prior", "working"), ...)
```

Arguments

formula an object of class "<u>formula</u>" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under Details.

122	R-GUI (R-Graphical User Interface)
family	a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See <u>family</u> for details of family functions.)
data	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in <code>data</code> , the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>glm</code> is called.
weights	an optional vector of weights to be used in the fitting process. Should be ${\tt NULL}$ or a numeric vector.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the <code>na.action</code> setting of <code>options</code> , and is <u><code>na.fail</code> if that is unset. The "factory-fresh" default is <u><code>na.omit</code>. Another possible value is <code>NULL</code>, no action. Value <u><code>na.exclude</code> can be useful.</u></u></u>
start	starting values for the parameters in the linear predictor.
etastart	starting values for the linear predictor.
mustart	starting values for the vector of means.
offset	this can be used to specify an <i>a priori</i> known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length either one or equal to the number of cases. One or more <u>offset</u> terms can be included in the formula instead or as well, and if both are specified their sum is used. See <u>model.offset</u> .
control	a list of parameters for controlling the fitting process. See the documentation for <u>glm.control</u> for details.
model	a logical value indicating whether <i>model frame</i> should be included as a component of the returned value.
method	the method to be used in fitting the model. The default method "glm.fit" uses iteratively reweighted least squares (IWLS). The only current alternative is "model.frame" which returns the model frame and does no fitting.
х, у	For glm: logical values indicating whether the response vector and model matrix used in the fitting process should be returned as components of the returned value. For glm.fit: x is a design matrix of dimension $n * p$, and y is a vector of observations of length p
contrasts	an optional list. See the contrasts.arg of model.matrix.default.

2006

object an object inheriting from class "glm".
type character, partial matching allowed. Type of weights to extract from the fitted
model object.
intercept logical. Should an intercept be included in the *null* model?
... further arguments passed to or from other methods.

Details

A typical predictor has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. For binomial and quasibinomial families the response can also be specified as a factor (when the first level denotes failure and all others success) or as a two-column matrix with the columns giving the numbers of successes and failures. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed. The terms in the formula will be reordered so that main effects come first, followed by the interactions, all second-order, all third-order and so on: to avoid this pass a terms object as the formula.

A specification of the form first:second indicates the the set of terms obtained by taking the interactions of all terms in first with all terms in second. The specification first*second indicates the *cross* of first and second. This is the same as first + second + first:second.

 ${\tt glm.fit}$ is the workhorse function.

If more than one of <code>etastart</code>, <code>start</code> and <code>mustart</code> is specified, the first in the list will be used. It is often advisable to supply starting values for a <code>guasi</code> family, and also for families with unusual links such as <code>gaussian("log")</code>.

All of weights, subset, offset, etastart and mustart are evaluated in the same way as variables in formula, that is first in data and then in the environment of formula.

Value

I Made Tirta

glm returns an object of class inheriting from "glm" which inherits from the class "lm". See later in this section. The function summary (i.e., <u>summary.glm</u>) can be used to obtain or print a summary of the results and the function <u>anova</u> (i.e., <u>anova.glm</u>) to produce an analysis of variance table. The generic accessor functions <u>coefficients</u>, effects, fitted.values and residuals can be used to extract various useful features of the value returned by glm. weights extracts a vector of weights, one for each case in the fit (after subsetting and na.action). An object of class "glm" is a list containing at least the following components:

coefficients	a named vector of coefficients
residuals	the <i>working</i> residuals, that is the residuals in the final iteration of the IWLS fit. Since cases with zero weights are omitted, their working residuals are ${\tt NA}.$
fitted.values	the fitted mean values, obtained by transforming the linear predictors by the inverse of the link function.
rank	the numeric rank of the fitted linear model.
family	the <u>family</u> object used.
linear.predictors	the linear fit on link scale.
deviance	up to a constant, minus twice the maximized log-likelihood. Where sensible, the constant is chosen so that a saturated model has deviance zero.
aic	Akaike's <i>An Information Criterion</i> , minus twice the maximized log- likelihood plus twice the number of coefficients (so assuming that the dispersion is known).
null.deviance	The deviance for the null model, comparable with deviance. The null model will include the offset, and an intercept if there is one in the model. Note that this will be incorrect if the link function depends on the data other than through the fitted mean: specify a zero offset to force a correct calculation.
iter	the number of iterations of IWLS used.
weights	the $\ensuremath{\textit{working}}$ weights, that is the weights in the final iteration of the IWLS fit.
prior.weights	the case weights initially supplied.
df.residual	the residual degrees of freedom.
df.null	the residual degrees of freedom for the null model.
У	the $_{\rm Y}$ vector used. (It is a vector even for a binomial model.)
converged	logical. Was the IWLS algorithm judged to have converged?

2006

boundary	logical. Is the fitted value on the boundary of the attainable values?
call	the matched call.
formula	the formula supplied.
terms	the terms object used.
data	the data argument.
offset	the offset vector used.
control	the value of the control argument used.
method	the name of the fitter function used, currently always "glm.fit".
contrasts	(where relevant) the contrasts used.
xlevels	(where relevant) a record of the levels of the factors used in fitting.

In addition, non-empty fits will have components qr, R and effects relating to the final weighted linear fit.

Objects of class "glm" are normally of class c("glm", "lm"), that is inherit from class "lm", and well-designed methods for class "lm" will be applied to the weighted linear model at the final iteration of IWLS. However, care is needed, as extractor functions for class "glm" such as <u>residuals</u> and weights do **not** just pick out the component of the fit with the same name. If a <u>binomial</u> glm model was specified by giving a two-column response, the weights returned by prior.weights are the total numbers of cases (factored by the supplied case weights) and the component y of the result is the proportion of successes.

Author(s)

The original ${\tt R}$ implementation of glm was written by Simon Davies working for Ross Ihaka at the University of Auckland, but has since been extensively re-written by members of the R Core team.

The design was inspired by the S function of the same name described in Hastie & Pregibon (1992).

References

I Made Tirta

Dobson, A. J. (1990) *An Introduction to Generalized Linear Models.* London: Chapman and Hall.

Hastie, T. J. and Pregibon, D. (1992) *Generalized linear models.* Chapter 6 of *Statistical Models in S* eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.

McCullagh P. and Nelder, J. A. (1989) *Generalized Linear Models.* London: Chapman and Hall.

Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S.* New York: Springer.

See Also

<u>anova.glm</u>, <u>summary.glm</u>, etc. for glm methods, and the generic functions <u>anova</u>, summary, <u>effects</u>, <u>fitted.values</u>, and <u>residuals</u>.

1m for non-generalized *linear* models (which SAS calls GLMs, for 'general' linear models).

 $\underline{\texttt{loglin}}$ and loglm for fitting log-linear models (which binomial and Poisson GLMs are) to contingency tables.

bigglm in package **biglm** for an alternative way to fit GLMs to large datasets (especially those with many cases).

esoph, infert and predict.glm have examples of fitting binomial glms.

Examples

```
## Dobson (1990) Page 93: Randomized Controlled Trial :
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
print(d.AD <- data.frame(treatment, outcome, counts))
glm.D93 <- glm(counts ~ outcome + treatment, family=poisson())
anova(glm.D93)
summary(glm.D93)
```

an example with offsets from Venables & Ripley (2002, p.189)

2006

I Made Tirta

2006

I Made Tirta

2006
INDEKS PENULIS

D

Dalgaard, 21, 26

F

Fox, 70, 79, 108

G

Grosjean, 11, 17, 22, 50, 63, 68

R

RDCT, 1, 15, 17, 31

S Schnute, 102

T Tirta, 1, 3, 1, 17, 92

Ζ

Zhang, 68,

I Made Tirta

2006

INDEKS SUBJEK

В

Bingkai, 6, 10, 51, 53

С

CLI, 7, 2, 3, 10, 97, 113

D

daftar, 23, 44, 45, 46 Diagram Pencar, 2, 104

Ε

Editor, 5, 6, 8, 11, 3, 10, 11, 12, 71, 73, 75, 77, 78, 80

F

Frame, 9, 11, 24, 52, 53, 66, 93, 107

G

grafik, 11, 1, 2, 3, 16, 71, 80, 81, 83, 85, 86, 87, 89, 91, 92, 99, 103 GUI, 1, 3, 5, 6, 9, 2, 3, 5, 8, 10,

16, 17, 19, 21, 22, 54, 63, 70, 97, 98, 101, 102, 108, 113, 114, 116

Η

html, 11, 76, 77, 78, 98, 116

Κ

komputer, 9, 7, 14, 16, 58, 60

L

Legal, 14 library, 5, 15, 27, 63, 64, 106 Linux, 8, 1, 7, 8, 15

Μ

Menu, 5, 6, 7, 8, 9, 10, 11, 3, 4, 5, 7, 8, 17, 18, 54, 55, 56, 57, 58, 73, 76, 78, 80, 97, 99, 108, 109, 113, 115

Ρ

package, 5, 103, 106, 116 paket, 1, 3, 5, 8, 10, 15, 16, 22, 76, 80, 89, 90, 95, 96, 97, 98, 99, 101, 102, 105, 108, 109, 112, 114, 115 pustaka, 5, 21, 22, 27, 28, 63, 110

Q

QQ-Plot, 2

2006

Laboratorium Statistika-Jurusan Matematika FMIPA UNEJ

Menu Berbahasa Indonesia siap dipanggil

55.

Rcmdr, 5, 7, 5, 16, 90, 108, 110, 111, 112, 114, 116 rgl, 11, 12, 89, 91, 96 RGUI, 5, 15, 16, 21, 22, 53, 63

S

R

Skrip, 5, 6, 8, 11, 10, 12, 31, 38, 71, 73, 74, 75, 78, 79, 97

Т

Tcl/Tk, 16, 101, 116 tcltk, 8, 16, 22, 27, 28, 31, 38, 40, 42, 45, 46, 48, 52, 54, 56, 59, 60, 61, 63, 64, 71, 73, 80, 87, 98, 99, 114, 116 tkrplot, 6, 11, 8, 16, 80, 81, 86, 87, 89, 114 TkRplot, 3, 70 tombol, 11, 22, 23, 24, 32, 33, 36, 38, 39, 40, 42, 44, 47, 48, 50, 52, 63, 64, 65, 66, 70, 71, 76, 80, 82, 85, 90, 91, 99, 102, 106, 107, 115

W

widget, 6, 21, 22, 24, 26, 27, 40, 51, 52, 58, 63, 64, 68, 70, 73, 80, 82, 83, 85, 86, 87, 93, 97, 99, 102, 103, 105, 113 Windows, 5, 8, 1, 3, 5, 7, 8, 13, 15, 116

I Made Tirta

2006