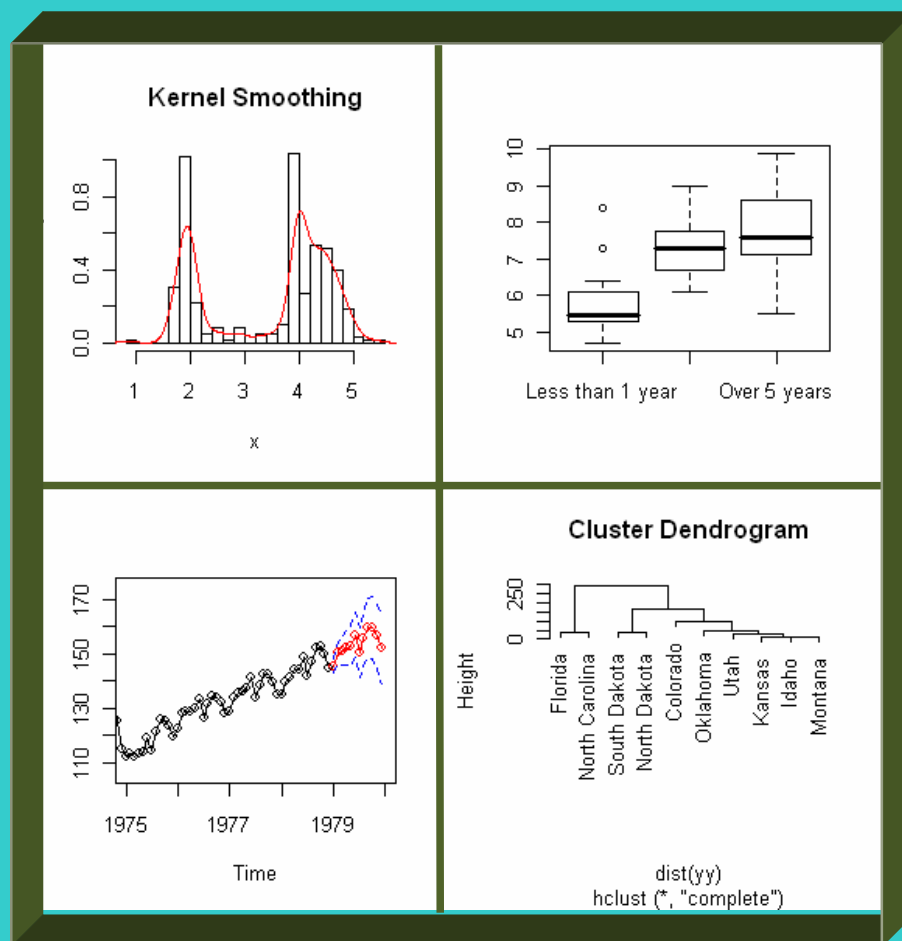


SUHARTONO

© 2008 Lab. Statistik Komputasi, ITS, Surabaya

ANALISIS DATA STATISTIK DENGAN *R*



When the Lord created the world and people to live in it — an enterprise which, according to modern science, took a very long time — I could well imagine that He reasoned with Himself as follows: “If I make everything predictable, these human beings, whom I have endowed with pretty good brains, will undoubtedly learn to predict everything, and they will thereupon have no motive to do anything at all, because they will recognize that the future is totally determined and cannot be influenced by any human action. On the other hand, if I make everything unpredictable, they will gradually discover that there is no rational basis for any decision whatsoever and, as in the first case, they will thereupon have no motive to do anything at all. Neither scheme would make sense. I must therefore create a mixture of the two. Let some things be predictable and let others be unpredictable. They will then, amongst many other things, have the very important task of finding out which is which.”

Small Is Beautiful

E. F. SCHUMACHER

*Untuk
Azizah, Alivia, Vanessa*

KATA PENGANTAR

R adalah suatu sistem untuk analisis data yang termasuk kelompok *software* statistik *open source* yang tidak memerlukan lisensi atau gratis, yang dikenal dengan *freeware*. Sampai saat ini, pengguna statistika di Indonesia masih belum banyak yang menggunakan **R** untuk keperluan analisis data. Sebagian besar pengguna statistika di Indonesia masih menggunakan paket-paket statistik komersil, seperti **SPSS**, **MINITAB**, **S-plus**, **SAS**, atau **Eviews**. Salah satu faktor penyebabnya adalah masih terbatasnya buku tentang **R** yang diterbitkan dalam bahasa Indonesia.

Buku ini bukan merupakan suatu buku teks tentang teori-teori dalam analisis statistik, tetapi lebih merupakan buku terapan tentang metode-metode statistik dengan penggunaan **R**. Tujuan penulisan buku ini adalah untuk menunjukkan bagaimana cara melakukan analisis data statistik dengan menggunakan **R**. Dalam hal ini, ditunjukkan bagaimana **R** sebagai suatu paket statistik yang *powerful* dan menyediakan sistem grafik yang baik untuk mendukung analisis. Jika proses perhitungan dalam analisis data menjadi mudah, maka energi dari pengguna statistika diharapkan dapat lebih difokuskan pada pemahaman tentang data yang dianalisis.

Buku ini ditujukan untuk pengguna **R** secara umum sebagai petunjuk pengantar pemakaian **R** untuk analisis data statistik. Selain itu, buku ini juga diharapkan dapat dipakai di kelas-kelas pada pengajaran statistika baik di level dasar ataupun level lanjut dengan teknik-teknik analisis statistik tertentu. Saat ini buku ini digunakan sebagai salah satu referensi pada mata kuliah Analisis Data I dan II di Program Sarjana (S1) dan mata kuliah Analisis Data di Program Magister (S2) Jurusan Statistika, Institut Teknologi Sepuluh Nopember (ITS), Surabaya.

Paket **R** memiliki fasilitas yang sangat banyak untuk analisis data statistik, mulai dari metode yang klasik sampai dengan yang modern. Pada Bab 1 diuraikan tentang paket statistik **R**, yaitu tentang sejarah singkat, cara memperoleh dan menginstal, serta fasilitas **R-GUI** (*Graphical User Interface*) atau **R-Commander** dan cara menginstalnya. Bab 2 dan 3 membahas tentang manajemen data di **R**, khususnya dengan menggunakan fasilitas di **R-Commander** dan perintah langsung di **R-Console**.

Analisis grafik pada **R** dijelaskan pada Bab 4, khususnya penggunaan fasilitas di **R-Commander**. Pada Bab 5 dibahas tentang penggunaan fasilitas di **R-Commander** untuk perhitungan fungsi distribusi peluang, yang mencakup perhitungan peluang pada distribusi kontinu dan diskrit. Bahasan tentang analisis statistik deskriptif dijelaskan pada Bab 6. Pada Bab 7 dijelaskan tentang penggunaan fasilitas di **R-Commander** untuk analisis statistik inferensi, yang mencakup uji hipotesis tentang rata-rata, proporsi, dan varians. Dalam Bab 8 dibahas tentang analisis regresi linear. Pada bagian akhir dari bab ini diberikan ringkasan beberapa perintah dan *library* yang berkaitan dengan analisis regresi.

Bahasan tentang penggunaan fasilitas di **R-Commander** untuk model linear tergeneralisir (GLM) dijelaskan pada Bab 9. Dalam Bab 10 dibahas tentang analisis grafik dengan menggunakan perintah langsung di **R-Console** atau **command line**. Bab 11 membahas tentang penggunaan **R** untuk analisis runtun waktu. Dalam bab ini ada tiga sub-bab utama tentang model-model dalam analisis runtun waktu yang dibahas, yaitu model tren linear, model eksponensial smoothing, dan model ARIMA. Di akhir bab ini diberikan pula ringkasan beberapa perintah dan *library* yang berkaitan dengan analisis runtun waktu.

Pada Bab 12 dijelaskan tentang penggunaan **R** untuk analisis multivariat, yang mencakup tentang Analisis Faktor, Analisis Diskriminan, dan Analisis Cluster. Dalam Bab 13 dijelaskan tentang model regresi nonparametrik dan estimasi densitas. Fokus pembahasan adalah pada regresi dengan *kernel* dan *spline*. Di akhir bab ini juga diberikan ringkasan beberapa perintah dan *library* yang berkaitan dengan aplikasi *kernel* dan *spline*.

Selanjutnya, pada Bab 14 dibahas tentang model non-linear. Dalam bab ini juga dibahas tentang beberapa uji statistik untuk deteksi hubungan non-linear, yaitu Uji Ramsey's RESET, Uji White, dan Uji Terasvirta. Pada akhirnya, dalam Bab 15 dijelaskan tentang pengantar pemrograman di **R**.

Pada kesempatan ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada dosen-dosen penulis yang telah banyak menginspirasi perkembangan akademik penulis, khususnya **Drs. Kresnayana Yahya, M.Sc.** dan **Ir. Dwiatmono A.W., M.lkom.** selama penulis menempuh S1 di ITS Surabaya, **Prof. T. Subba Rao** dan **Dr. Jingsong Yuan** dari *Department of Mathematics, University of Manchester, United Kingdom*, selama penulis menempuh S2, dan **Prof. Subanar, Ph.D.** selama penulis menempuh S3 di UGM Yogyakarta. Penulis juga mengucapkan banyak terima kasih kepada kolega-kolega akademik penulis yang telah banyak membantu dalam proses penulisan buku ini, khususnya **R. Mohamad Atok, S.Si., M.Si.** dan **Wahyu Wibowo, S.Si., M.Si.** Akhirnya, penulis juga mengucapkan banyak terima kasih kepada mahasiswa/i penulis, khususnya mahasiswa/i S1 Statistika 2005 yang telah melakukan *download* paket dan *library R* secara bersama-sama sehingga banyak *library* (hampir 1000 *library*) yang sekarang telah tersedia dan dapat diaktifkan.

Masukan dan umpan balik dari pembaca sangat diharapkan untuk perbaikan isi buku ini. Pembaca dapat mengirimkan saran dan kritik melalui email ke alamat penulis, yaitu suhartono@statistika.its.ac.id atau har_arema@yahoo.com. Semoga buku ini dapat memberikan manfaat, khususnya bagi perkembangan ilmu statistika di Indonesia dan secara umum bagi para pembaca.

Surabaya, 29 September 2008
Penulis,
Suhartono

DAFTAR ISI

	hal.
KATA PENGANTAR	iv
DAFTAR ISI	vi
BAB 1. PAKET STATISTIK R	1
1.1 Pendahuluan	1
1.2 Sejarah Singkat R	1
1.3 Cara Memperoleh R, Paket dan Library	1
1.4 Instalasi R dalam Sistem Operasi Windows	2
1.5 GUI R-Commander dan Instalasinya dalam Sistem Operasi Windows	4
1.6 Manajemen Direktori Kerja di R	6
1.7 Fasilitas help	10
1.7.1 Mencari help dari suatu perintah (command) tertentu	10
1.7.2 Menggunakan help-search-engine	12
1.7.3 Online Search-Engine	15
BAB 2. MANAJEMEN DATA DI PAKET R	16
2.1 Data Entry menggunakan R-Gui dengan R-Commander	16
2.2 Menampilkan data yang sedang aktif di R-Commander	19
2.3 Editing data di R-Commander	20
2.4 Importing data di R-Commander	20
2.4.1 Importing data file Excel di R-Commander	20
2.4.2 Importing data file SPSS di R-Commander	22
2.4.3 Importing data file MINITAB di R-Commander	22
2.5 Memilih dataset yang aktif	24
2.6 Transformasi dataset atau pengaturan variabel pada dataset	25
2.6.1 Recode atau kode ulang peubah	25
2.6.2 Compute atau hitung peubah baru	27
BAB 3. MANAJEMEN DATA DI R DENGAN COMMAND LINE	29
3.1 Jenis-jenis Data Objek	30
3.1.1 Data Array Satu Dimensi atau Data Vektor	30
3.1.2 Data Matriks	31
3.1.3 Data Frame	34
3.1.4 Data List	37
3.2 Importing Data pada Command Line	38
3.2.1 Membaca File ASCII	38
3.2.2 Importing Data File Excel	39
3.2.3 Importing Data dari Paket Statistik	41

BAB 4. GRAFIK MENGGUNAKAN R-Commander	43
4.1 Grafik dalam R-GUI	45
4.2 Grafik Histogram	46
4.3 Diagram Dahan dan Daun (Stem-and-Leaf)	48
4.4 Grafik Box-Plot	50
4.5 Grafik QQ-Plot	51
4.6 Grafik Diagram Pencar (Scatter-Plot)	53
4.7 Grafik Plot Rata-rata (Mean)	55
4.8 Diagram Batang (Bar-Chart)	56
4.9 Diagram Lingkaran (Pie-Chart)	57
4.10 Plot Indeks	58
BAB 5. FUNGSI DISTRIBUSI PELUANG DI R-Commander	61
5.1 Fungsi Distribusi Kontinu	62
5.1.1 Menghitung Kuantil dari Distribusi Normal	62
5.1.2 Menghitung Peluang dari Distribusi Normal	64
5.1.3 Membuat Plot dari Distribusi Normal	65
5.1.4 Membangkitkan Data dari Distribusi Normal	67
5.2 Fungsi Distribusi Diskrit	70
5.2.1 Menghitung Kuantil dari Distribusi Binomial	71
5.2.2 Menghitung Peluang dari Distribusi Binomial	72
5.2.3 Membuat Plot dari Distribusi Binomial	74
5.2.4 Membangkitkan Data dari Distribusi Binomial	76
BAB 6. STATISTIK DESKRIPTIF MENGGUNAKAN R-Commander	80
6.1 Ringkasan Numerik (Summary)	81
6.1.1 Ringkasan Numerik dari Semua Variabel	81
6.1.2 Ringkasan Numerik untuk Suatu Variabel	83
6.2 Distribusi Frekuensi	85
6.3 Tabel Statistika	86
6.4 Matriks Korelasi	88
6.5 Uji Korelasi	89
6.6 Uji Kenormalan Shapiro-Wilk	91
6.7 Tabel Kontingensi Dua Arah	92
6.8 Entry Langsung Data Frekuensi untuk Tabel Kotingensi Dua Arah	94
BAB 7. STATISTIK INFERENSI MENGGUNAKAN R-Commander	97
7.1 Pengujian Rata-rata (Mean)	99
7.1.1 Pengujian Rata-rata sampel tunggal	99
7.1.2 Pengujian Perbedaan Rata-rata Dua sampel saling bebas	102
7.1.3 Pengujian Perbedaan Rata-rata Sampel Berpasangan	107

7.1.4 Analisis Varians satu arah (One-way ANOVA)	110
7.1.5 Analisis Varians dua arah (Multi-way ANOVA)	115
7.2 Pengujian Kesamaan Variansi	118
7.2.1 Pengujian Kesamaan Dua Variansi	118
7.2.2 Uji Bartlett	120
7.2.3 Uji Levene	121
7.3 Pengujian Proporsi	122
7.3.1 Pengujian Proporsi Sampel Tunggal	123
7.3.2 Pengujian Proporsi Dua Sampel	125
BAB 8. ANALISIS REGRESI MENGGUNAKAN R-Commander	128
8.1 Regresi Linear	128
8.2 Model Linear	132
8.3 Cek Diagnosa Kesesuaian Model Regresi Linear	137
8.4 Rangkuman perintah dan library yang berkaitan dengan Analisis Regresi	144
BAB 9. GENERALIZED LINEAR MODEL MENGGUNAKAN R-Commander	158
9.1 Pengantar Teori Model Linear Tergeneralisir	158
9.2 Contoh Kasus Model Linear Tergeneralisir dengan R-Commander	161
BAB 10. GRAFIK MENGGUNAKAN R-CLI	166
10.1 Fungsi-fungsi Plot Utama	168
10.1.1 Perintah plot()	168
10.1.2 Perintah qqnorm(x), qqline(x), qqplot(x,y)	172
10.1.3 Perintah hist(x)	176
10.1.4 Perintah image(x,y,z,...), contour(x,y,z,...), persp(x,y,z,...)	177
10.1.5 Argumen-argumen untuk fungsi plot utama	178
10.2 Fungsi-fungsi Plot Tambahan	179
10.3 Fungsi-fungsi Plot yang bersifat interaktif	180
10.4 Notasi Matematika pada Plot	180
10.5 Setting parameter grafik	182
BAB 11. ANALISIS RUNTUN WAKTU DENGAN R	184
11.1 Model Trend Linear	185
11.2. Model Exponential Smoothing	187
11.2.1 Model Holt-Winters Aditif	189
11.2.2 Model Holt-Winters Multiplikatif	193
11.2.3 Model Eksponensial Ganda	195
11.2.4 Model Eksponensial Smoothing Sederhana	196
11.3 Model ARIMA	198
11.3.1 Contoh Kasus Model ARIMA Non-musiman yang Stasioner	203

11.3.2	Contoh Kasus Model Non-musiman yang Tidak Stasioner	211
11.3.3	Model ARIMA Musiman	216
11.3.4	Contoh Kasus Model ARIMA Musiman	219
11.3.5	Kriteria Pemilihan Model	225
11.4	Rangkuman perintah dan library yang berkaitan dengan Analisis Runtun Waktu	227
BAB 12.	ANALISIS MULTIVARIAT DENGAN R	230
12.1	Analisis Faktor	230
12.2	Analisis Diskriminan	232
12.3	Analisis Cluster	234
BAB 13.	REGRESI NONPARAMETRIK DAN ESTIMASI DENSITAS	237
13.1	Estimasi Densitas dengan Kernel	237
13.2	Regresi Nonparametrik dengan Kernel	241
13.3	Regresi Nonparametrik dengan Spline	243
13.4	Jenis-jenis Basis Spline	249
13.5	Rangkuman library untuk Aplikasi Kernel dan Spline	254
BAB 14.	MODEL NON-LINEAR	256
14.1	Estimasi Model Regresi Non-linear	256
14.2	Perintah nls dan SSasympOrig untuk estimasi model non-linear	259
14.3	Uji Deteksi Hubungan Non-linear	263
14.3.1	Uji Ramsey's RESET	263
14.3.2	Uji White	267
14.3.3	Uji Terasvirta	274
BAB 15.	PENGENALAN PEMROGRAMAN DALAM R	277
15.1	Penulisan Fungsi	277
15.2	Type Data dan Operator	280
15.3	Control Flow di dalam R	281
15.4	Beberapa topik yang berhubungan dengan fungsi	284
15.4.1	Argumen dari suatu fungsi	284
15.4.2	Mengatur tampilan dari output	286
15.5	Contoh-contoh fungsi	289
	DAFTAR PUSTAKA	292
	DAFTAR INDEKS	295
	TENTANG PENULIS	298

BAB 1

PAKET STATISTIK R

1.1. Pendahuluan

Secara umum ada dua macam kelompok paket *software* statistik untuk keperluan analisis data, yaitu kelompok *software* komersil dan kelompok *software* statistik *open source* atau *freeware*. Beberapa contoh *software* statistik komersil yang populer di Indonesia adalah SPSS, MINITAB, Eviews, SAS, dan Splus. Sedangkan contoh dari *freeware* statistik antara lain R, Open Stats, SalStat, Vista, dan lain-lain (lihat <http://www.statistics.com/content/freesoft/AZlisting.html>).

Software statistik yang komersil mensyaratkan lisensi dengan harga yang relatif sangat mahal untuk ukuran sebagian besar pengguna di Indonesia. Dengan demikian, salah satu alternatif penyelesaian dari mahalnya lisensi tersebut adalah melalui penggunaan *freeware* statistik, khususnya R.

1.2. Sejarah Singkat R

R dalam versi terakhirnya, yaitu versi 2.7.2 per 25 Agustus 2008, merupakan suatu sistem analisis data statistik yang komplet sebagai hasil dari kolaborasi penelitian berbagai ahli statistik (statistisi) di seluruh dunia. Versi awal dari R dibuat pada tahun 1992 di Universitas Auckland, New Zealand oleh Ross Ihaka dan Robert Gentleman. Pada saat ini, *source code kernel* R dikembangkan terutama oleh **R Core Team** yang beranggotakan 17 orang statistisi dari berbagai penjuru dunia (lihat <http://www.r-project.org/contributors.html>). Selain itu, para statistisi lain pengguna R di seluruh dunia juga memberikan kontribusi berupa kode, melaporkan bug, dan membuat dokumentasi untuk R.

Paket statistik R bersifat *multiplatforms*, dengan file instalasi *binary/file tar* tersedia untuk sistem operasi Windows, Mac OS, Mac OS X, Linux, Free BSD, NetBSD, irix, Solaris, AIX, dan HP-UX. Secara umum, sintaks dari bahasa R adalah ekuivalen dengan paket statistik Splus, sehingga sebagian besar keperluan analisis statistika, dan pemrograman dengan R adalah hampir identik dengan perintah yang dikenal di Splus.

1.3. Cara Memperoleh R, Paket dan Library

R dapat diperoleh secara gratis di *CRAN-archive* yaitu *The Comprehensive R Archive Network* di alamat <http://cran.r-project.org>. Pada server CRAN ini dapat didownload file instalasi *binary* dan *source code* dari *R-base system* dalam sistem operasi Windows (semua versi), beberapa jenis distro linux, dan Macintosh.

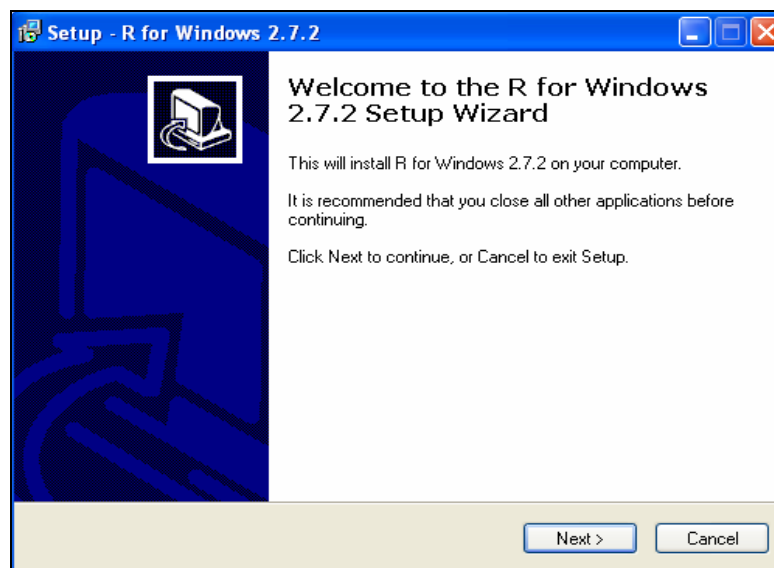
Fungsi dan kemampuan dari R sebagian besar dapat diperoleh melalui *Add-on packages/library*. Suatu *library* adalah kumpulan perintah atau fungsi yang dapat di-

gunakan untuk melakukan analisis tertentu. Sebagai contoh, fungsi untuk melakukan analisis *time series* dapat diperoleh di *library ts*. Instalasi standar dari R akan memuat berbagai *library* dasar, antara lain *base*, *datasets*, *graphics*, *utils*, dan *stats*. *Library* lain hasil kontribusi dari pengguna R (di luar yang standar) harus diinstal satu per satu sesuai dengan yang dibutuhkan untuk analisis. Daftar semua *library* yang tersedia dapat diakses dari link download CRAN di alamat <http://cran.r-project.org>.

1.4. Instalasi R dalam Sistem Operasi Windows

Tahapan utama sebelum melakukan instalasi R dalam sistem operasi Windows adalah mendownload file **R-2.7.2-win32.exe** yang dapat diperoleh di <http://cran.r-project.org>. Setelah itu, langkah-langkah instalasi R dapat dilakukan seperti berikut:

- Klik dua kali (*double click*) file **R-2.7.2-win32.exe** yang terdapat pada direktori yang telah disediakan, maka akan muncul jendela dialog seperti pada Gambar 1.1. berikut ini.



Gambar 1.1. Jendela dialog awal instalasi R dalam sistem operasi Windows

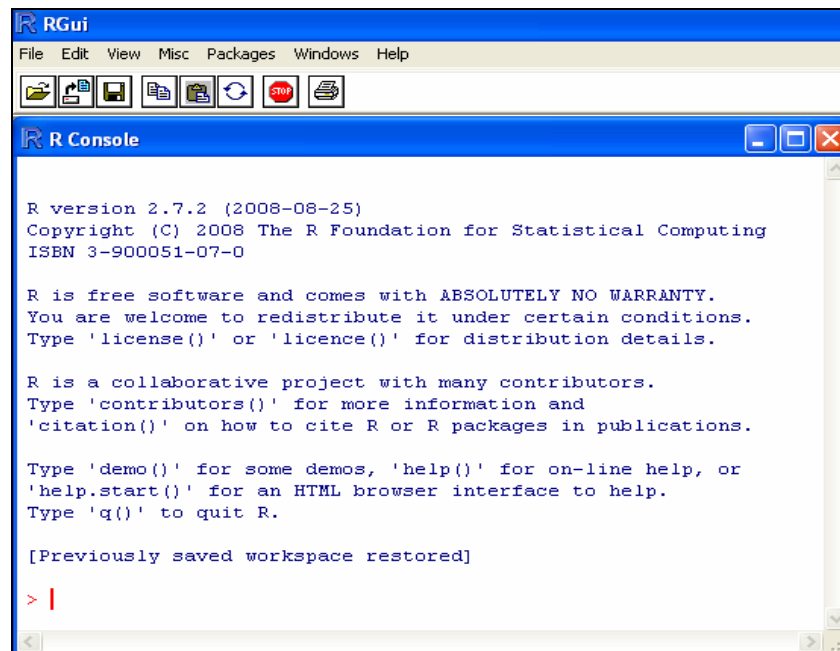
Setelah itu, lanjutkan jalannya proses instalasi dengan mengikuti *Wizard* dan menggunakan pilihan-pilihan *default* instalasi.

- Jika proses instalasi telah selesai, klik **Finish** untuk keluar dari proses instalasi. Apabila semua proses berjalan dengan sukses, maka pada **Desktop Windows** dan **Start Menu** dari Windows akan terdapat **Shortcut** dari R seperti pada Gambar 1.2. berikut ini.



Gambar 1.2. Shortcut dari R

- Langkah terakhir jika instalasi R telah selesai adalah melakukan pengecekan atau pengujian apakah program R dapat berjalan dengan baik. Lakukan klik dua kali pada **shortcut** R di **Desktop** atau pada **Start Menu**. Jika instalasi berlangsung dengan baik, maka jendela program R akan terbuka seperti yang terlihat pada Gambar 1.3.



Gambar 1.3. Jendela awal program R, jika instalasi berjalan sukses

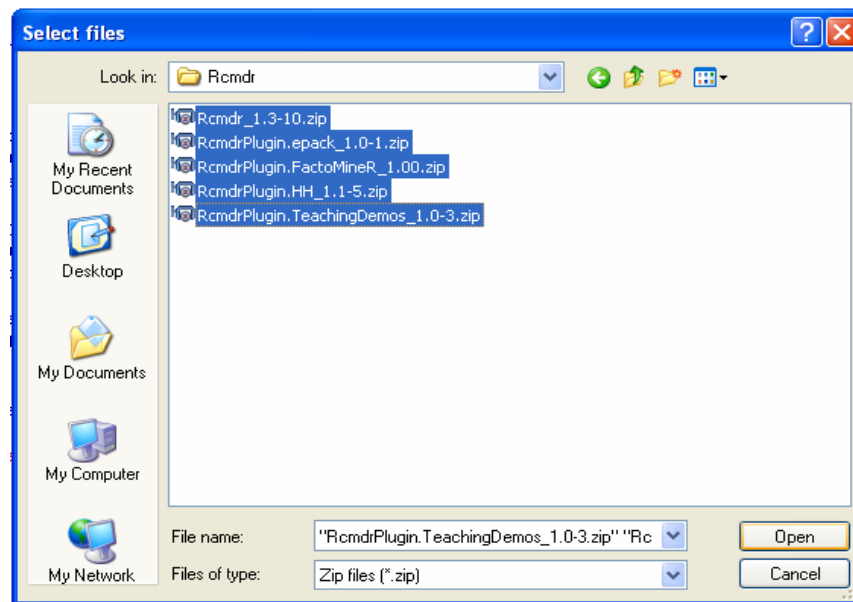
- Jika selesai bekerja dengan R, maka untuk keluar dari R dapat dilakukan dengan dua cara, yaitu :
 1. Ketikkan **q()** pada **command line** di **R-console**, yaitu
`> q()`
 2. Pilih menu **File**, pilih **Exit**, dan kemudian klik **Yes** pada dialog **Save Workspace Image**.

1.5. GUI R-Commander dan Instalasinya dalam Sistem Operasi Windows

Pada awalnya, interaksi utama antara pengguna dengan R adalah bersifat *Command Line Interface* (CLI). Dengan demikian, untuk dapat menggunakan R diperlukan penyesuaian-penyesuaian bagi pengguna yang telah terbiasa dengan fasilitas *Point and Click Graphical User Interface* (GUI). Untungnya, pada saat ini telah tersedia beberapa GUI sederhana untuk keperluan beberapa analisis statistika tertentu, khususnya yang berkaitan dengan manajemen data di R. *Library R-commander* yang terdiri dari **Rcmdr**, **RcmdrPlugin.TechingDemos**, **RcmdrPlugin.epack**, **RcmdrPlugin.HH**, dan **RcmdrPlugin.FactoMineR**, merupakan *library* tambahan dari R untuk memfasilitasi GUI yang dapat digunakan untuk berbagai analisis statistika dasar.

Instalasi *library R-commander* dapat dilakukan apabila file-file *library* di atas sudah didownload dari server CRAN. Jika instalasi untuk R telah selesai dan berjalan sukses, maka langkah-langkah untuk instalasi *R-commander* adalah sebagai berikut:

1. Pertama, jalankan program R sampai jendela program R terbuka (seperti yang terlihat pada Gambar 1.3 di atas).
2. Untuk menginstal *R-commander*, pilih menu **Packages**, pilih **Install package(s) from local zip files ...**. Kemudian arahkan lokasi pada dialog **Look in** ke direktori dimana file *Rcmdr_1.3-10.zip*, *RcmdrPlugin.HH_1.1-5.zip*, *RcmdrPlugin.epack_1.0-1.zip*, *RcmdrPlugin.TechingDemos_1.3-10.zip*, *RcmdrPlugin.FactoMineR_1.00.zip*. Pilih semua file tersebut, seperti yang terlihat pada jendela dialog pada Gambar 1.4.



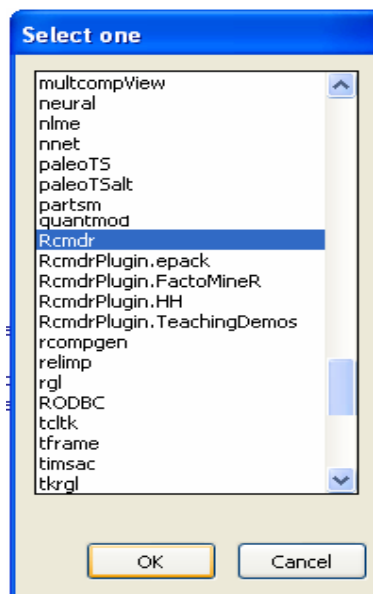
Gambar 1.4. Jendela dialog untuk instalasi *R-commander*

Kemudian klik **Open**, maka R akan menginstal paket *R-commander* yang ditandai dengan dialog berikut pada jendela **R-console**.

```
> utils::menuInstallLocal()
package 'RcmdrPlugin.TeachingDemos' successfully unpacked and MD5 sums
checked
package 'Rcmdr' successfully unpacked and MD5 sums checked
package 'RcmdrPlugin.epack' successfully unpacked and MD5 sums checked
package 'RcmdrPlugin.FactoMineR' successfully unpacked and MD5 sums
checked
package 'RcmdrPlugin.HH' successfully unpacked and MD5 sums checked
updating HTML package descriptions
>
```

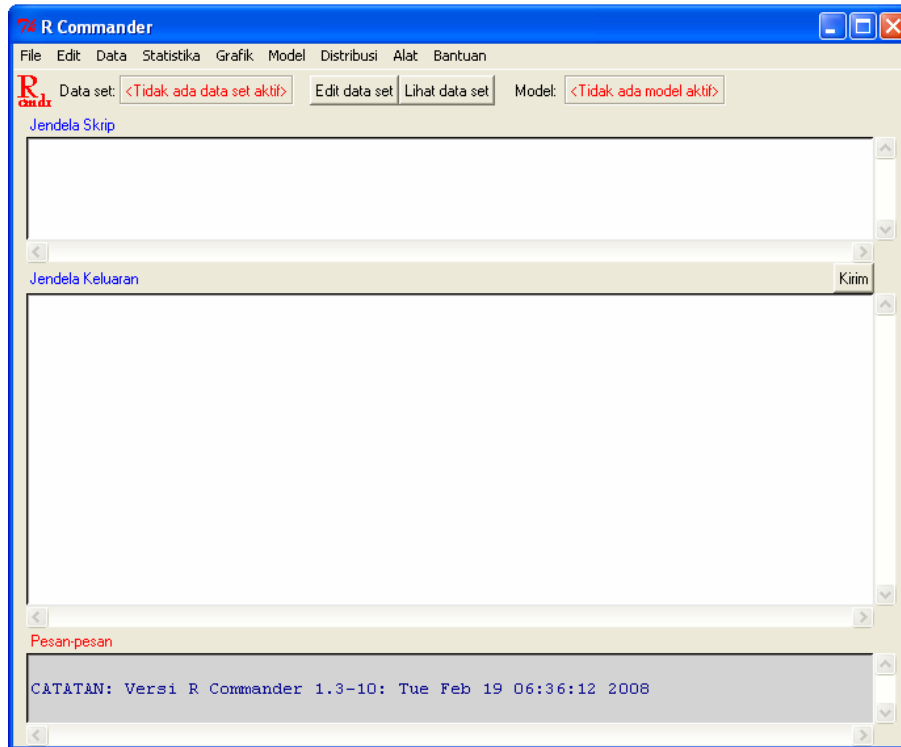
3. Paket *R-commander* dapat dijalankan dengan dua cara yang berbeda, yaitu:

- Dengan mengetikkan perintah **library(Rcmdr)** pada jendela **R-console** dan menekan **Enter** satu kali.
 > library(Rcmdr)
- Memilih menu **Packages**, pilih **Load package ...** dan kemudian memilih **Rcmdr** pada daftar paket *library* yang telah terinstal, seperti yang terlihat pada Gambar 1.5.



Gambar 1.5. Jendela dialog untuk menjalankan *R-commander*

Apabila proses instalasi paket *R-commander* berjalan dengan sukses, maka paket *R-commander* tersebut akan diloading dan muncul seperti pada Gambar 1.6 berikut ini.



Gambar 1.6. Jendela awal dari paket *library R-commander*

Pada saat ini, bahasa yang digunakan dalam paket *R-commander* sudah ada yang dalam bahasa Indonesia sebagai hasil pengembangan dan kontribusi statistisi di Indonesia.

4. Untuk keluar dari paket *R-commander* dan sekaligus R dapat dilakukan dengan memilih menu **File**, pilih **Keluar**, dan klik pada pilihan **Dari Commander dan R** yang tersedia di jendela *R-commander*.

1.6. Manajemen Direktori Kerja di R

Cara kerja dari R adalah sama dengan Splus, yaitu bekerja dengan satu direktori untuk satu proyek. R akan menyimpan file *image* dari semua obyek atau *internal data* dan *history* dari semua perintah yang pernah diketikkan di jendela **R-console** pada direktori kerja secara otomatis atau *default* dengan file berekstensi **.Rdata**. Lokasi

default dari direktori kerja R adalah direktori “C:\Program Files\R\R-2.7.2”. Untuk keperluan pekerjaan sehari-hari yang menggunakan R akan lebih baik jika dilakukan pada direktori tersendiri, misalnya direktori dengan nama yang sesuai dengan pekerjaan yang dijalankan. Dengan demikian akan memudahkan dalam melihat *history* dan *obyek* yang berhubungan dengan pekerjaan tersebut.

1.6.1. Mengubah lokasi direktori kerja atau workspace

Berikut ini adalah langkah-langkah yang dapat digunakan untuk membuat direktori khusus dari suatu pekerjaan dengan menggunakan R.

- Misalkan kita telah mempunyai direktori **C:\Kerja_dg_R** (buatlah direktori ini jika belum ada). Langkah pertama, buatlah satu direktori baru di **C:\Kerja_dg_R** dengan nama direktori “**Nama_Pekerjaan**”, misalkan **Kerja1**. Dengan demikian, pada tahap ini diperoleh suatu direktori baru yaitu

C:\Kerja_dg_R\Kerja1.

- Buatlah copy dari *shortcut* program R di **desktop window**, dan *rename shortcut* ini sebagai *shortcut Kerja1*. Sehingga di **desktop window** muncul *shortcut Kerja1* seperti Gambar 1.7 berikut ini.

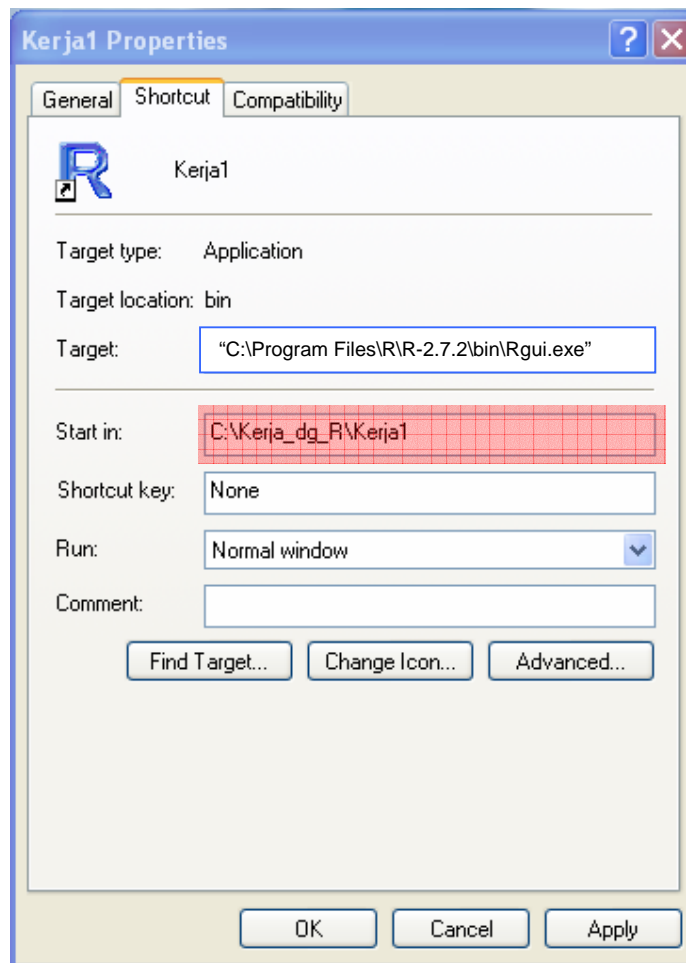


Gambar 1.7. *Shortcut* di **desktop window** dengan nama **Kerja1**

- Kemudian arahkan *mouse* pada *shortcut* tersebut dan klik kanan. Pilih **Properties** dan ganti informasi pada kolom **Start in** menjadi **C:\Kerja_dg_R\Kerja1** seperti yang terlihat pada Gambar 1.8, setelah itu klik **OK**.

Untuk mengetahui perubahan lokasi direktori kerja di R, lakukan klik dua kali pada *icon shortcut Kerja1* untuk menjalankan R. Sebagai ilustrasi sederhana, ketikkan beberapa baris perintah berikut ini setelah jendela R terbuka.

```
> x=1:15
> y=x+5
> x
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
> y
[1] 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```



Gambar 1.8. Perubahan lokasi direktori kerja ke C:\Kerja_dg_R\Kerja1

Setelah mengetikkan beberapa baris perintah di atas, lakukan keluar dari R dengan memilih menu **File/Exit**. Pada dialog pertanyaan **Save workspace image?**, klik pada pilihan **YES**. Sekarang, jika dilakukan *browsing* di direktori **C:\Kerja_dg_R\Kerja1** maka akan ditemukan satu file bernama **.Rdata** yang merupakan nama *default* file image dari direktori kerja, dan file yang lain bernama **.Rhistory** yang merupakan nama *default* dari file yang berisikan *history* dari semua perintah yang pernah diketikkan. Kedua file ini secara *default* akan diloading oleh **R** pada saat dijalankan untuk suatu sesi pekerjaan. *History* dan data dari suatu sesi terakhir (yang telah tersimpan sebelum keluar) dapat diakses pada jendela **R-console** dengan menggunakan tanda panah ke atas dan ke bawah.

1.6.2. Menyimpan image direktori kerja

Pada bagian sebelumnya telah dijelaskan bagaimana semua obyek yang digunakan dalam satu sesi pemakaian R, yaitu mulai dibukanya program R sampai ditutup kembali, akan disimpan secara *default* ke dalam file **.Rdata**. Supaya file-file pekerjaan lebih terorganisir, R memberikan fasilitas tambahan untuk menyimpan data atau obyek yang digunakan dalam setiap sesi R ke dalam file *workspace* tertentu yang memiliki ekstensi **.Rdata**. File-file ini selanjutnya dapat di-load kembali jika diperlukan. Sebagai contoh, jalankan program R dan ketikkan beberapa perintah berikut ini ke dalam jendela **R-console**.

```
> x=1:15
> y=x+5
> x
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
> y
[1] 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

Proses penyimpanan data atau obyek, yaitu x dan y seperti yang tertulis di atas, ke dalam direktori **C:\Kerja_dg_R\Kerja1** dengan nama file **coba1.Rdata** dapat dilakukan dengan menggunakan menu **File**, dan pilih **Save Workspace ...**. Selanjutnya lakukan keluar dari R, dan pilih **No** (yang berarti tidak menyimpan image dari file kerja) pada dialog **Save Workspace Image?** Sekarang jalankan kembali program R, maka data atau obyek di file **coba1.Rdata** dapat di-load kembali dengan menggunakan dua macam cara, yaitu:

- Pilih menu **File**, dan pilih **Load Workspace ...**, dan setelah itu pilih file di direktori **C:\Kerja_dg_R\Kerja1** dengan nama **coba1.Rdata**
- Gunakan perintah berikut pada jendela **R-console**

```
> load("C:\\Kerja_dg_R\\Kerja1\\Coba1.RData")
> objects()
[1] "x" "y"
> x
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
> y
[1] 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

Dari jendela kotak **R-console** di atas dapat dilihat bahwa semua obyek yang dikerjakan pada sesi sebelumnya telah berhasil di-load kembali. Dengan cara yang sama, semua *history* dari perintah pada suatu sesi dapat disimpan melalui menu **File**, dan pilih **Save**

History... . Untuk melakukan load kembali *history* pada sesi sebelumnya yang sudah tersimpan ini, dapat dilakukan dengan melalui menu **File**, dan pilih **Load History ...** , kemudian pilih nama file *history* yang akan dipanggil kembali tersebut. *History* dari sesi R yang telah diload ini dapat diakses dengan menggunakan tanda panah ke atas dan ke bawah.

1.7. Fasilitas help

Secara umum ada beberapa fasilitas *help* dari R yang dapat diakses dengan berbagai cara, antara lain:

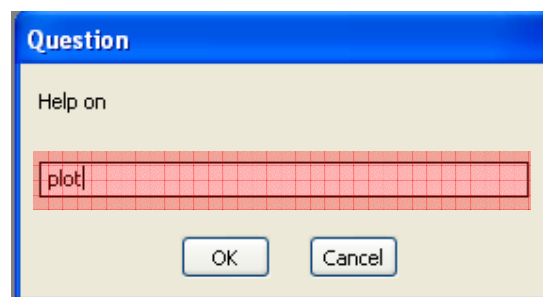
- Mencari help dari suatu perintah (command) tertentu
- Menggunakan help-search-engine
- Online Search-Engine

1.7.1. Mencari help dari suatu perintah (command) tertentu

Ada beberapa perintah yang dapat digunakan untuk mencari *help* atau bantuan terhadap suatu fungsi atau perintah dari R yang telah diketahui namanya. Sebagai contoh, jika ingin diketahui secara detail tentang suatu perintah atau fungsi R yang bernama “**plot**”, maka pada jendela **R-console** dapat diketikkan salah satu perintah berikut ini, yaitu:

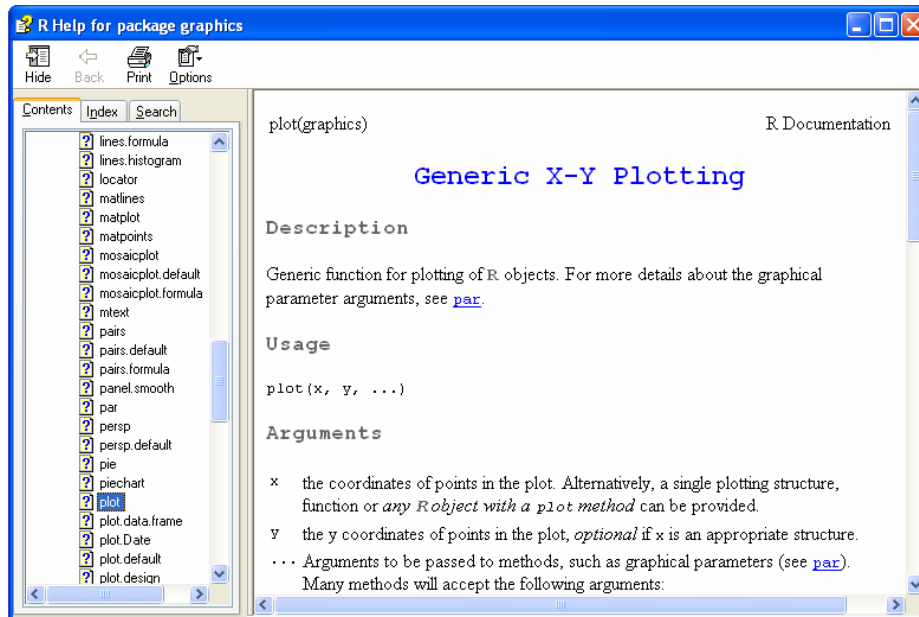
```
> help(plot)
> ?plot
```

Jendela *help* yang sama dapat juga diperoleh dengan menggunakan menu dengan pilihan **Help**, dan pilih **R function (text) ...** dan setelah jendela dialog muncul, ketikkan kata **plot** seperti yang terlihat pada Gambar 1.9 berikut ini.



Gambar 1.9. Jendela dialog *help* untuk suatu fungsi atau perintah

Setelah salah satu dari perintah di atas dijalankan, maka akan ditampilkan bagian dari jendela help dari perintah plot seperti yang terlihat pada Gambar 1.10 berikut ini.



Gambar 1.10. Hasil pencarian *help* untuk suatu fungsi **plot**

Penjelasan dari jendela hasil pencarian help untuk fungsi plot ini adalah sebagai berikut:

- Ada dua kolom jendela yang muncul, yaitu kolom kiri tentang **index** dari fungsi atau perintah yang dicari (misal **plot**), dan kolom kanan adalah hasil atau penjelasan dari pencarian fungsi yang ingin diketahui.
- Pada bagian kiri atas kolom jendela hasil *help* adalah tentang keterangan nama dari perintah atau fungsi yang sedang ditampilkan dan nama paket atau *library* yang memuat perintah tersebut. Dalam contoh di atas, untuk perintah **plot** dapat dilihat bahwa perintah **plot** ini tersimpan dalam paket atau *library graphics*..
- Pada setiap jendela help dari suatu perintah secara umum akan memuat bagian-bagian berikut:
 - **Description**: uraian singkat tentang perintah tersebut
 - **Usage**: uraian tentang syntax perintah untuk penggunaan perintah tersebut.
 - **Arguments**: uraian tentang argumen-argumen yang diperlukan dari fungsi atau perintah tersebut.
 - **Details**: uraian yang lebih lengkap (daripada yang diberikan pada bagian description) tentang perintah tersebut.

- **Values:** uraian tentang output perintah tersebut.
- **Author(s):** uraian tentang author dari perintah tersebut.
- **References:** uraian tentang referensi yang dapat digunakan untuk memperoleh keterangan lebih lanjut dari perintah tersebut.
- **See also:** bagian ini berisi daftar perintah atau fungsi yang berkaitan erat dengan perintah tersebut.
- **Example:** bagian ini berisi contoh-contoh penggunaan perintah tersebut.

1.7.2. Menggunakan help-search-engine

Metode pencarian help lain yang dapat dilakukan adalah dengan menggunakan pencarian terhadap “kata kunci”. Beberapa metode yang dapat dilakukan untuk tujuan ini dapat dijelaskan seperti berikut ini.

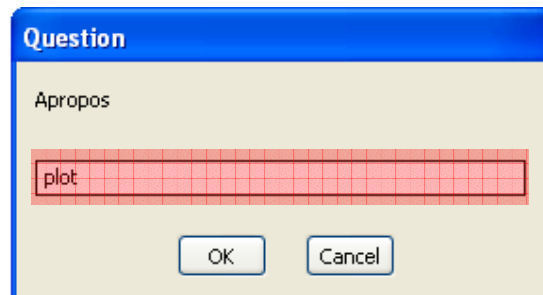
a. Menggunakan perintah apropos(“kata kunci”)

Perintah ini dapat digunakan untuk memperoleh daftar perintah-perintah dari semua paket atau *library* yang telah terinstal pada sistem R yang memuat suatu “kata kunci”. Berikut ini adalah contoh hasil perintah **apropos(plot)**.

```
> apropos("plot")
[1] ". _C _recordedplot"      "assocplot"          "barplot"
[4] "barplot.default"        "biplot"              "boxplot"
[7] "boxplot.default"        "boxplot.stats"      "cdplot"
[10] "coplot"                  "fourfoldplot"        "interaction.plot"
[13] "lag.plot"                "matplot"              "monthplot"
[16] "mosaicplot"              "plot"                 "plot.default"
[19] "plot.density"            "plot.design"          "plot.ecdf"
[22] "plot.lm"                 "plot.mlm"             "plot.new"
[25] "plot.spec"               "plot.spec.coherency"  "plot.spec.phase"
[28] "plot.stepfun"            "plot.ts"              "plot.TukeyHSD"
[31] "plot.window"             "plot.xy"              "preplot"
[34] "qqplot"                  "recordPlot"           "replayPlot"
[37] "savePlot"                "screeplot"            "spineplot"
[40] "sunflowerplot"          "termplot"             "ts.plot"
>
```

Dari hasil jendela di **R-console** tersebut dapat dilihat output yang memuat kata kunci “**plot**” dalam suatu nama perintah. Output yang diperoleh akan berbeda dan tergantung pada *library* yang terinstal pada komputer.

Output yang sama dapat pula diperoleh dengan menggunakan menu utama pada pilihan **Help**, kemudian pilih **Apropos ...** dan selanjutnya ketik **plot** pada jendela dialog seperti yang terlihat pada Gambar 1.11 berikut ini.



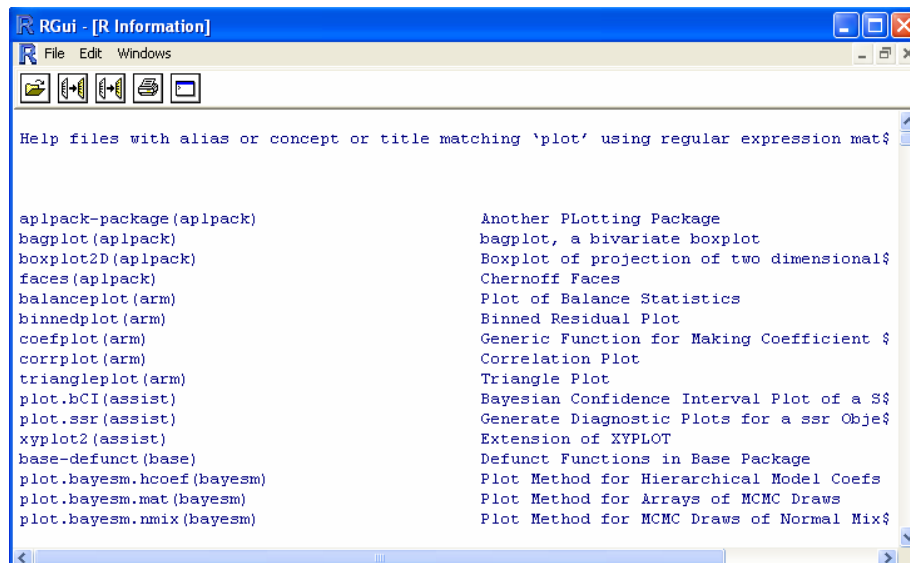
Gambar 1.11. Jendela dialog *Apropos* untuk pencarian suatu perintah

b. Menggunakan perintah `help.search("kata kunci")`

Perintah ini akan melakukan pencarian terhadap sebuah *string* bernama kata kunci di semua paket atau *library* yang telah terinstal pada sistem R. Berikut ini adalah contoh perintah `help.search("plot")` pada jendela **R-console**.

```
> help.search("plot")
```

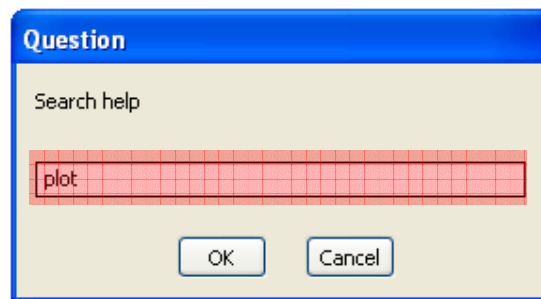
Hasil dari perintah tersebut dapat dilihat pada Gambar 1.12 berikut ini.



Gambar 1.12. Hasil pencarian *help* dengan perintah `help.search("plot")`

Dari hasil untuk contoh di atas dapat dilihat keterangan nama perintah atau fungsi beserta nama paket atau *library* (kata yang didalam kurung) yang memuat *string* “**plot**”. Output yang diperoleh akan berbeda dan tergantung pada library yang terinstal pada komputer.

Output yang sama dapat pula diperoleh dengan menggunakan menu utama pada pilihan **Help**, kemudian pilih **Search help** dan selanjutnya ketik **plot** pada jendela dialog seperti yang terlihat pada Gambar 1.13 berikut ini.



Gambar 1.13. Jendela dialog *Search help* untuk pencarian suatu perintah

c. Menggunakan versi html dari jendela help

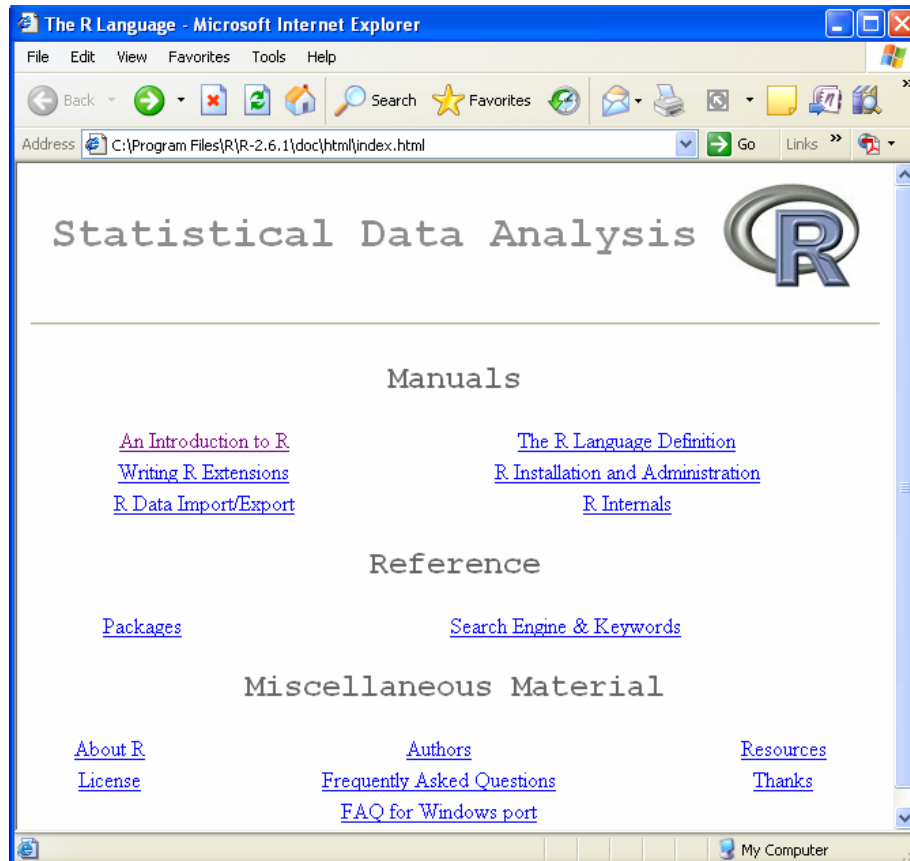
Tampilan dari *help* dalam versi **html** dapat diperoleh melalui fungsi atau perintah **help.start()** pada jendela R-console. Selain itu, jendela help dalam html ini dapat pula dibuka menggunakan menu pada pilihan **Help**, dan kemudian pilih **Html help**. Berikut ini adalah contoh **help.start()** pada jendela **R-console**.

```
> help.start()
```

Hasil dari perintah tersebut dapat dilihat pada Gambar 1.14. Beberapa keterangan atau uraian dari hasil jendela help versi **html** adalah sebagai berikut:

- Pada bagian **Manuals**, diperoleh daftar *link* dari semua file manual dalam versi **html** dari R. Versi file **pdf** dari file manual ini dapat diakses melalui menu utama **Help**, dan pilih **Manuals (in Pdf)** dari R.
- Pada bagian **Reference** terdiri dari dua informasi utama, yaitu tentang **Package** yang berisi daftar semua paket atau *library* yang telah diinstal pada sistem, dan tentang **Search Engine & Keywords** yang dapat digunakan untuk pencarian kata kunci (*keywords*) dalam semua paket atau *library* yang telah diinstal dalam sistem R yang ada di komputer.

- Pada bagian **Miscellaneous Material** terdiri dari beberapa link beberapa informasi tambahan yang penting untuk diketahui lebih lanjut.



Gambar 1.14. Hasil pencarian *help* dalam versi **html** dengan perintah **help.start()**

1.7.3. Online Search-Engine

Informasi tentang R secara *online* dapat dicari dengan menggunakan *search engine* di alamat **<http://cran.r-project.org/search.html>**. Pada alamat tersebut dapat diperoleh semua informasi tentang R yang ada dalam situs **CRAN**, informasi semua paket atau *library* yang tersedia untuk R, dan ditambah informasi yang tersedia pada *archive mailing list* **r-help@stat.math.ethz.ch**.

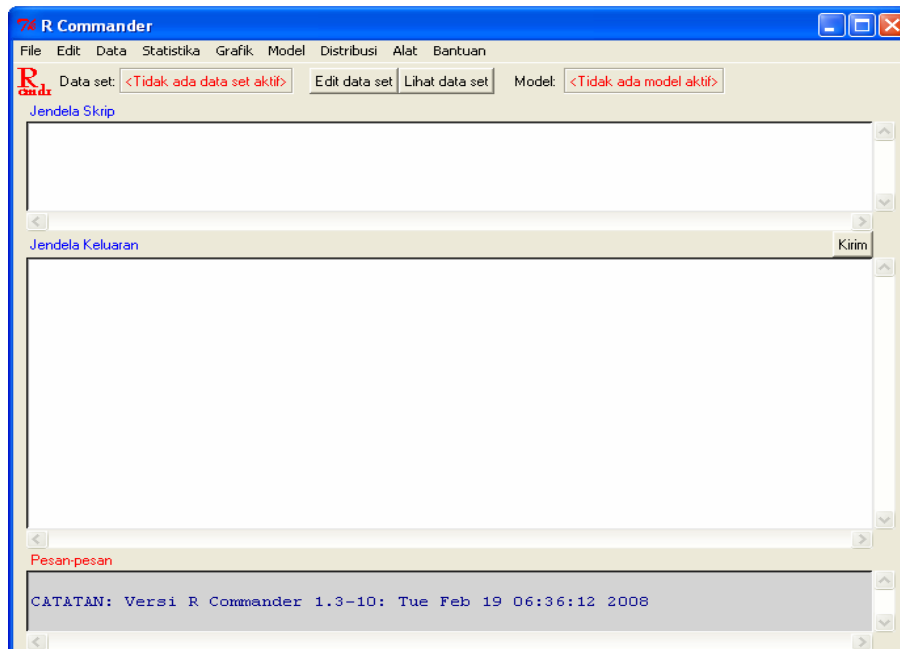
BAB 2

MANAJEMEN DATA DI PAKET R

Manajemen data yang meliputi *data entry, edit, import* dan *export*, merupakan suatu langkah yang penting dalam analisis statistika. Ada beberapa macam dan ukuran data yang dapat diolah menggunakan R. Secara umum, minimal ada dua macam bentuk data yang dapat diolah, yaitu data yang dimasukkan langsung lewat **R editor** melalui *keyboard*, dan data yang sudah ditulis menggunakan **Program Sheet** lain, seperti *Text, SPSS, MINITAB, Access* ataupun *dBase*. R menyediakan dua cara untuk melakukan manajemen data, yaitu menggunakan **R-GUI** dan melalui *command line* di **R-console**. Pada bab ini, pembahasan tentang manajemen data difokuskan yang melalui **R-GUI**, khususnya pemakaian **R-Commander**.

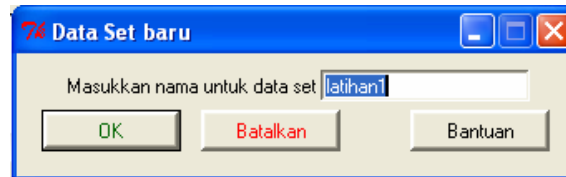
2.1. Data Entry menggunakan R-Gui dengan R-Commander

Pada tahap awal, aktifkan kembali program R dengan mengklik *icon shortcutnya*. Kemudian *load library R-Commander* dengan mengetikkan perintah **library(Rcmdr)** pada jendela **R-console**, dan tunggu sampai **R-Commander** selesai diloading. Jika proses berjalan sukses maka akan nampak jendela **R-Commander** seperti pada Gambar 2.1.



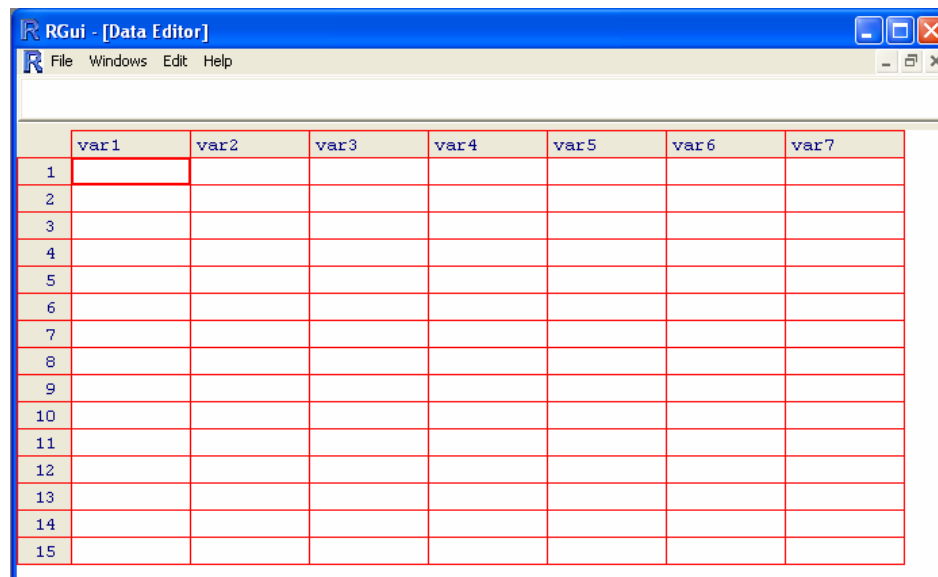
Gambar 2.1. Jendela awal dari paket *library R-commander* yang sukses diloading

Pengisian data secara langsung via **R** dengan menggunakan **R-commander** dapat dilakukan melalui menu **Data**, dan pilih **Dataset baru ...**. Setelah itu, jendela dialog pengisian nama data set akan ditampilkan, seperti yang terlihat pada Gambar 2.2. Pada kotak dialog nama data set, tuliskan **latihan1** sebagai nama data set baru tersebut.



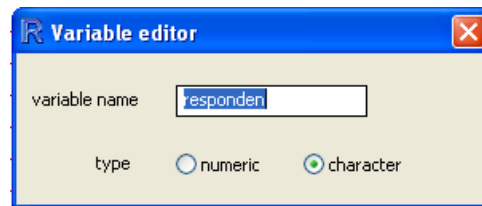
Gambar 2.2. Jendela dialog pengisian nama data set

Kemudian klik **OK**, dan jendela **RGui - Data Editor** akan terbuka seperti pada Gambar 2.3 berikut ini.



Gambar 2.3. Jendela **RGui - Data Editor** untuk pengisian data

Pengisian nama variabel dilakukan dengan cara klik pada kolom paling atas dari data editor. Sebagai contoh, untuk mengisi nama variabel pertama, misalnya responden, klik pada **var1**. Kemudian pada jendela **Variable editor** seperti yang terlihat pada Gambar 2.4, isikan **responden** sebagai **variable name** dan tipe data adalah **character** (karena yang akan diisi pada kolom ini adalah nama-nama responden).



Gambar 2.4. Jendela **Variable editor** untuk pengisian nama variabel

Sebagai latihan, isikan data tentang nama mahasiswa, nilai UAN tiga mata pelajaran, dan IPK semester 1, berikut ini kedalam **R Data editor**.

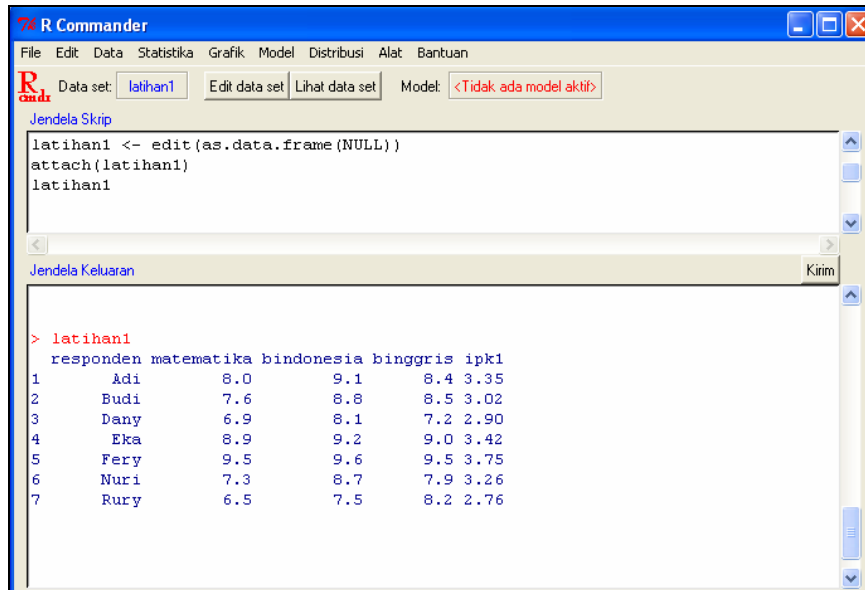
Responden	Matematika	Bindonesia	BInggris	IPK1
Adi	8.0	9.1	8.4	3.35
Budi	7.6	8.8	8.5	3.02
Dany	6.9	8.1	7.2	2.90
Eka	8.9	9.2	9.0	3.42
Fery	9.5	9.6	9.5	3.75
Nuri	7.3	8.7	7.9	3.26
Rury	6.5	7.5	8.2	2.76

Pada dasarnya, proses pengisian data ini adalah sama dengan paket statistik yang lain, yaitu mulai isian nama kolom dan tipe data yang diinputkan (**numeric** atau **character**). Setelah semua data selesai diinputkan, maka akan diperoleh tampilan Data Editor seperti berikut ini.

	responden	matematika	bindonesia	binggris	ipk1	var6
1	Adi	8	9.1	8.4	3.35	
2	Budi	7.6	8.8	8.5	3.02	
3	Dany	6.9	8.1	7.2	2.9	
4	Eka	8.9	9.2	9	3.42	
5	Fery	9.5	9.6	9.5	3.75	
6	Nuri	7.3	8.7	7.9	3.26	
7	Rury	6.5	7.5	8.2	2.76	
8						

Gambar 2.5. Jendela **Data Editor** setelah semua data selesai diisikan

Setelah dilakukan *data entry*, maka tutup jendela **R Data Editor** diatas untuk mengakhiri proses *data entry*. Pada jendela **R-Commander** terlihat **Data set** yang dengan nama **latihan1** saat ini sedang aktif, seperti yang terlihat pada Gambar 2.6. Untuk menampilkan data yang sedang aktif di **Jendela Keluaran R-Commander**, tulis nama data set yaitu **latihan1** di **Jendela Skrip**, kemudian klik **Kirim**, maka akan terlihat data seperti berikut ini.



Gambar 2.6. Jendela R Commander setelah dilakukan proses *entry data*

2.2. Menampilkan data yang sedang aktif di R-Commander

Untuk menampilkan data yang sedang aktif di memori, lakukan dengan mengklik tombol **Lihat data set**. Setelah itu jendela data akan dibuka dan menampilkan data yang sedang aktif di memori komputer saat ini, yaitu data **latihan1** berikut ini.

	responden	matematika	bindonesia	binggris	ipk1
1	Adi	8.0	9.1	8.4	3.35
2	Budi	7.6	8.8	8.5	3.02
3	Dany	6.9	8.1	7.2	2.90
4	Eka	8.9	9.2	9.0	3.42
5	Fery	9.5	9.6	9.5	3.75
6	Nuri	7.3	8.7	7.9	3.26
7	Rury	6.5	7.5	8.2	2.76

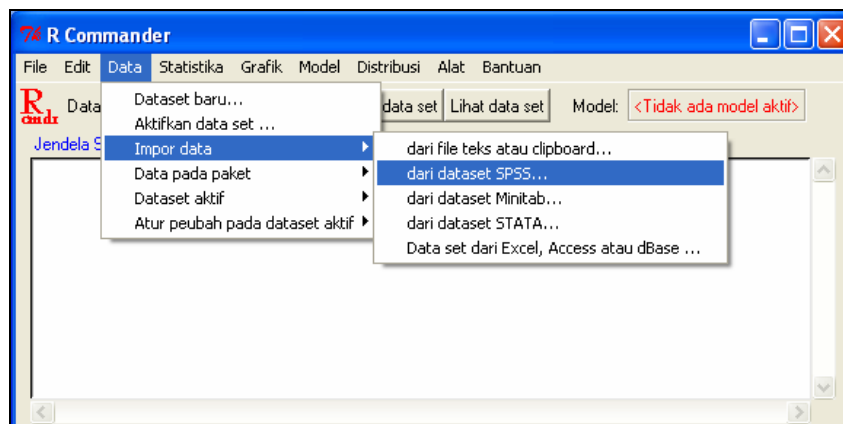
Gambar 2.7. Jendela data **latihan1** yang sedang aktif di memori

2.3. Editing data di R-Commander

Untuk melakukan editing terhadap data **latihan1**, lakukan dengan mengklik tombol **Edit data set**. Setelah itu jendela **Data Editor** akan dibuka kembali, dan proses editing data dapat langsung dilakukan pada data-data yang salah ketik. Jika editing telah selesai dilakukan, tutup jendela **Data Editor** untuk kembali ke jendela **R-commander**. Hasil editing yang telah dilakukan dapat dilihat dengan klik pada tombol **Lihat data set**.

2.4. Importing data di R-Commander

Seperti yang telah dijelaskan pada bagian sebelumnya, secara umum proses *data entry* di **R-Commander** dapat dilakukan dengan dua macam cara, yaitu dilakukan langsung melalui **Data Editor** dan melalui import data dari format data yang diberikan oleh program lain. Program yang format datanya dapat dibaca oleh **R** adalah data dari file **teks** atau **clipboard**, dataset **SPSS**, dataset **MINITAB**, dataset **STATA**, data dari **Excel**, **Access**, atau **dBase**, seperti yang terlihat pada jendela menu berikut.

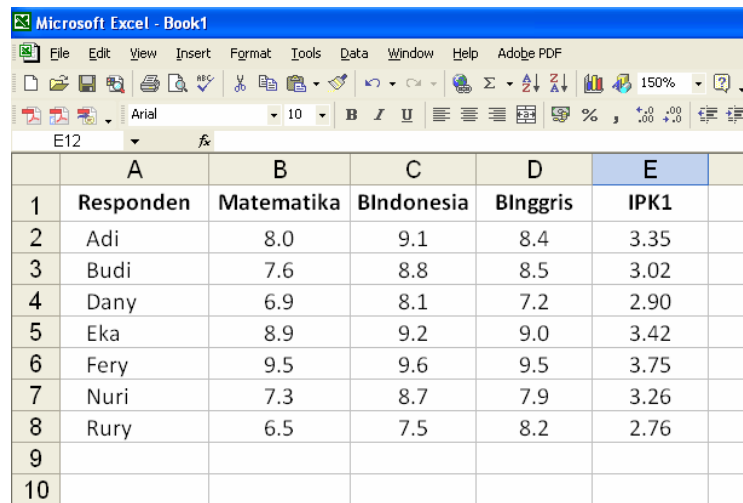


Gambar 2.8. Jendela Impor data pada R-Commander

Pada bagian berikut ini akan dijelaskan penggunaan impor data dari **Excel**, **SPSS**, dan **MINITAB**. Untuk file dari program yang lain, proses impor data melalui **R-Commander** dapat dilakukan secara sama dengan cara mengimpor data dari program **Excel**, **SPSS**, ataupun **MINITAB**.

2.4.1. Importing data file Excel di R-Commander

Misalkan saja data file **Excel** belum ada, dan akan dibuat terlebih dahulu. Buka program **Excel**, setelah itu isikan data tentang responden diatas sehingga diperoleh data **Excel** seperti yang terlihat pada Gambar 2.9.



	A	B	C	D	E
1	Responden	Matematika	Blndonesia	BInggris	IPK1
2	Adi	8.0	9.1	8.4	3.35
3	Budi	7.6	8.8	8.5	3.02
4	Dany	6.9	8.1	7.2	2.90
5	Eka	8.9	9.2	9.0	3.42
6	Fery	9.5	9.6	9.5	3.75
7	Nuri	7.3	8.7	7.9	3.26
8	Rury	6.5	7.5	8.2	2.76
9					
10					

Gambar 2.9. Jendela data pada Excel yang akan diimpor ke R

Langkah selanjutnya, simpan file ini sebagai file **text** (yaitu *tab delimited txt*), dengan nama **data1.txt** di direktori **C:\Kerja_R**. Untuk mengimpor data file ini kedalam **R-Commander**, pilihlah pada **R-Commander** menu **Data**, pilih **Impor data**, dan kemudian pilih **dari file teks atau clipboard ...**. Pada jendela dialog yang muncul, isikan informasi *nama untuk data set*, *nama variabel*, dan lain-lain, seperti berikut ini.



Baca Data Dari File Teks atau Clipbo...

Masukkan nama untuk data set: latihan2

Nama peubah dalam file: ☒

Baca data dari clipboard: ☐

Indikator data hilang: NA

Pemisah Field/Medan

Spasi: ☒

Koma: ☐

Tab: ☐

Yang lain: ☐ Spesifikasi:

Karakter Titik-Desimal

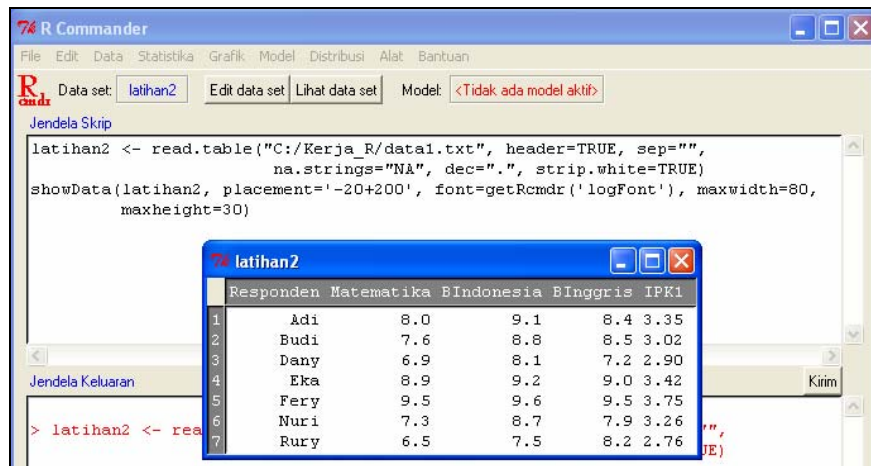
Titik [.]: ☒

Koma [,]: ☐

OK Batalkan Bantuan

Gambar 2.10. Jendela dialog Impor data dari file teks atau clipboard

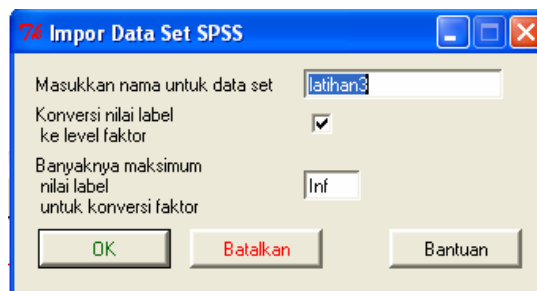
Dalam hal ini, data hasil impor akan disimpan kedalam **R-Commander** dengan nama **latihan2**. Data ini diimpor dengan **Pemisah Field/Medan** adalah spasi. Klik **OK**, kemudian akan muncul untuk melakukan browsing ke lokasi dari file teks yang akan diimpor. Arahkan ke direktori **C:\Kerja_R** dan pilih file **data1.txt**. Kemudian klik **Open**, maka sekarang data yang berada pada file **data1.txt** telah diimpor kedalam **R-Commander** dengan nama **latihan2**. Sekarang, data set yang aktif pada **R-Commander** adalah **latihan2** seperti yang terlihat pada Gambar 2.11. Gunakan tombol **Lihat data set** untuk melihat hasil impor data ini.



Gambar 2.11. Jendela dialog hasil impor data dan latihan2 sebagai data set aktif

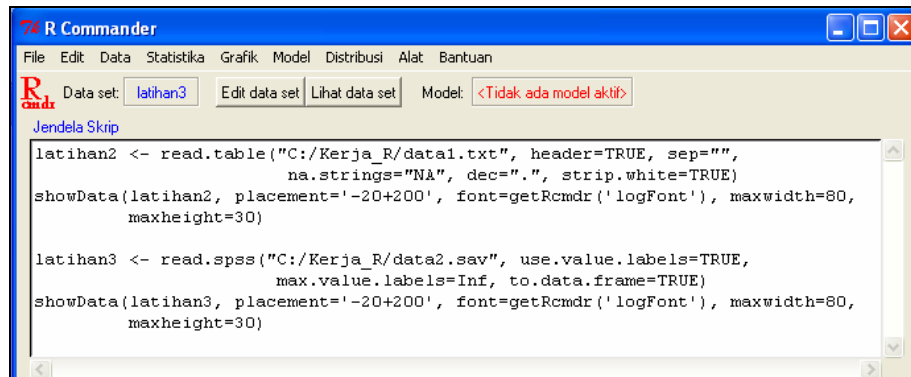
2.4.2. Importing data file SPSS di R-Commander

Proses impor data eksternal yang telah disimpan sebagai file SPSS, dapat dilakukan dengan memilih pada **R-Commander** menu **Data**, pilih **Impor data**, dan kemudian pilih **dari dataset SPSS...**. Pada jendela dialog yang muncul, isikan informasi *nama untuk data set* (misal **latihan3**) seperti berikut ini.



Gambar 2.12. Jendela dialog Impor Dataset SPSS

Klik **OK**, dan selanjutnya arahkan ke direktori tempat penyimpanan file SPSS yang akan diimpor, misalkan saja di **C:\Kerja_R** dengan nama **data2.sav**. Kemudian klik **Open**, maka data hasil impor dari file **data2.sav** akan disimpan kedalam file **latihan3**. Pada jendela **R-Commander** terlihat data set **latihan3** sedang aktif, seperti pada Gambar 2.13. Klik tombol **Lihat data set** untuk melihat hasil impor data ini.



Gambar 2.13. Jendela dialog hasil impor data dan **latihan3** sebagai data set aktif

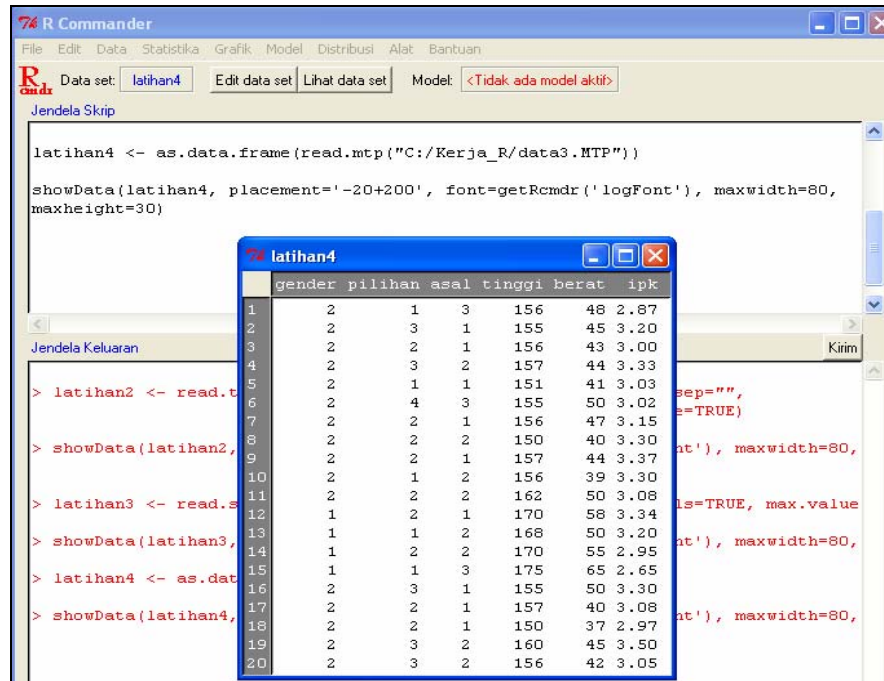
2.4.3. Importing data file MINITAB di R-Commander

Proses impor data eksternal yang telah disimpan sebagai file MINITAB adalah ekuivalen dengan impor data teks atau SPSS sebelumnya, yaitu dapat dilakukan dengan memilih pada **R-Commander** menu **Data**, pilih **Impor data**, dan kemudian pilih **dari dataset Minitab...**. Pada jendela dialog yang muncul, isikan informasi *nama untuk data set* (misal **latihan4**) seperti berikut ini.



Gambar 2.14. Jendela dialog **Impor Dataset MINITAB**

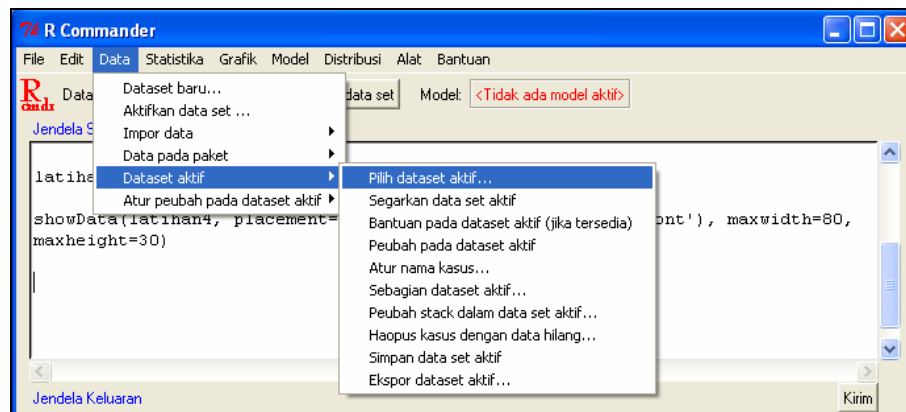
Klik **OK**, dan selanjutnya arahkan ke direktori tempat penyimpanan file MINITAB yang akan diimpor, misalkan saja di **C:\Kerja_R** dengan nama **data3.MTP** (*Minitab Portable Worksheet*, lakukan pada saat **save as** di Minitab). Kemudian klik **Open**, maka data hasil impor dari file **data3.MTP** akan disimpan kedalam file **latihan4**. Pada jendela **R-Commander** terlihat data set **latihan4** sedang aktif, seperti pada Gambar 2.15. Klik tombol **Lihat data set** untuk melihat hasil impor data ini.



Gambar 2.15. Jendela dialog hasil impor data dan latihan4 sebagai data set aktif

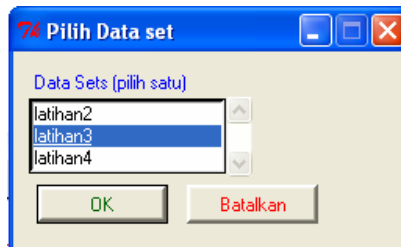
2.5. Memilih dataset yang aktif

Pemilihan dataset yang aktif pada **R-Commander** dapat dilakukan dengan menggunakan menu **Data**, pilih **Dataset aktif**, dan kemudian klik **Pilih dataset aktif...** seperti berikut ini.



Gambar 2.16. Jendela dialog untuk memilih menu dataset yang sedang aktif

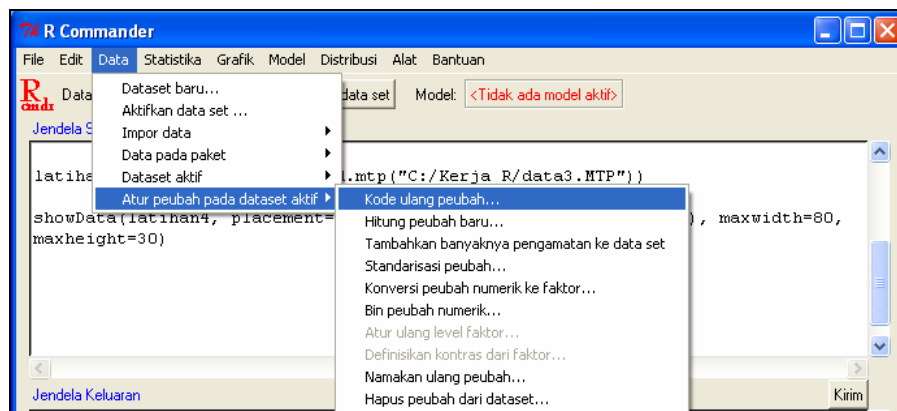
Selanjutnya, pilihlah dataset yang ingin diaktifkan dengan melakukan klik pada nama dataset yang dipilih, kemudian klik **OK** seperti berikut ini.



Gambar 2.17. Jendela dialog untuk memilih dataset yang sedang aktif

2.6. Transformasi dataset atau pengaturan variabel pada dataset

Ada beberapa menu untuk transformasi dataset pada **R-Commander**, antara lain **recode** atau **kode ulang peubah**, **compute** atau **hitung peubah baru**, standarisasi peubah, dan lainnya. Secara lengkap, transformasi dataset yang dapat dilakukan dapat dilihat pada Gambar 2.18.



Gambar 2.18. Beberapa menu untuk melakukan transformasi dataset

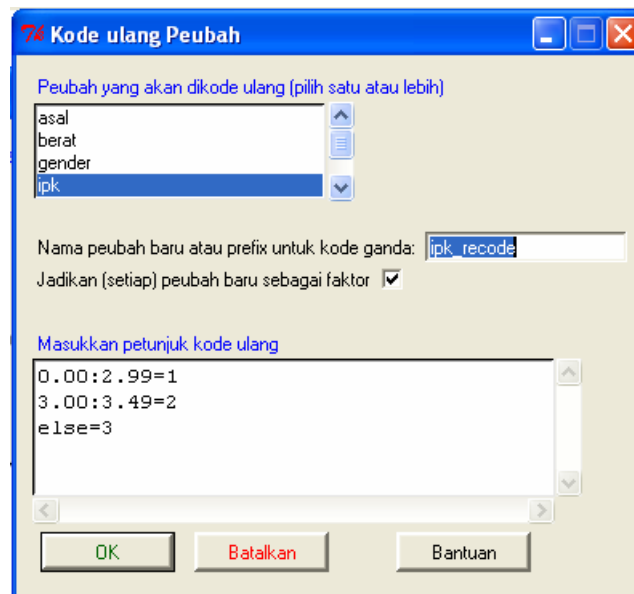
2.6.1. Recode atau kode ulang peubah

Kode ulang peubah merupakan pilihan menu pertama pada pengaturan variabel dataset. Misalkan saja akan dilakukan *recode* atau kode ulang pada variabel IPK dari dataset latihan4.

Range nilai IPK	Nilai kode baru
< 3.00	1
3.00 – 3.50	2
> 3.50	3

Langkah-langkah pengkodean dapat dilakukan sebagai berikut.

- Buka menu **recode** dengan memilih menu **Data** pada **R-Commander**, kemudian pilih **Atur peubah pada dataset aktif**, dan selanjutnya pilih **Kode ulang peubah ...**. Selanjutnya akan diperoleh tampilan seperti berikut.

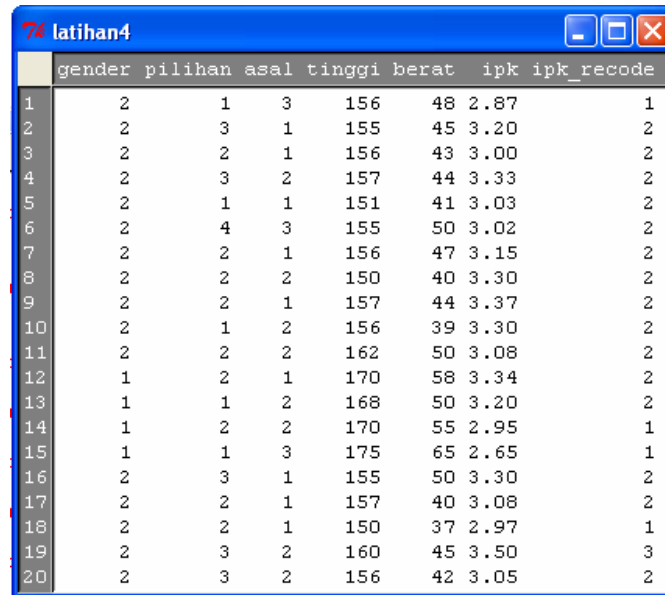


Gambar 2.19. Jendela pilihan Kode ulang Peubah

- Kemudian arahkan ke variabel **ipk**, dan namakan hasil recode sebagai **ipk_recode**. Informasi pengkodean dapat dijelaskan dengan menggunakan informasi berikut:

```
0.00:2.99 = 1
3.00:3.49 = 2
else      = 3
```

- Klik OK, dan sekarang jika dilihat pada dataset **latihan4**, akan diperoleh variabel baru yaitu **ipk_recode** yang berisikan data hasil pengkodean ulang dari ipk. Lakukan dengan klik **Lihat dataset**, sehingga diperoleh tampilan data seperti berikut.



	gender	pilihan	asal	tinggi	berat	ipk	ipk_recode
1	2	1	3	156	48	2.87	1
2	2	3	1	155	45	3.20	2
3	2	2	1	156	43	3.00	2
4	2	3	2	157	44	3.33	2
5	2	1	1	151	41	3.03	2
6	2	4	3	155	50	3.02	2
7	2	2	1	156	47	3.15	2
8	2	2	2	150	40	3.30	2
9	2	2	1	157	44	3.37	2
10	2	1	2	156	39	3.30	2
11	2	2	2	162	50	3.08	2
12	1	2	1	170	58	3.34	2
13	1	1	2	168	50	3.20	2
14	1	2	2	170	55	2.95	1
15	1	1	3	175	65	2.65	1
16	2	3	1	155	50	3.30	2
17	2	2	1	157	40	3.08	2
18	2	2	1	150	37	2.97	1
19	2	3	2	160	45	3.50	3
20	2	3	2	156	42	3.05	2

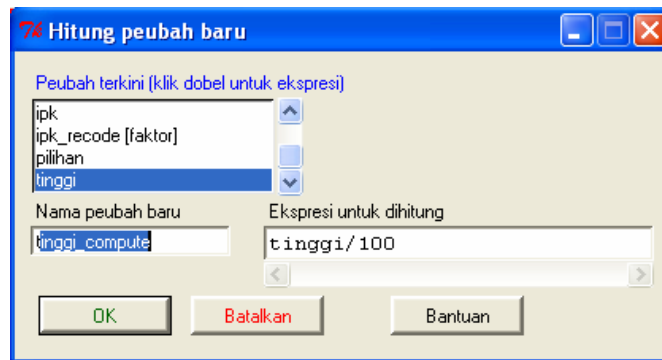
Gambar 2.20. Jendela data hasil Kode ulang Peubah **ipk** menjadi **ipk_recode**

2.6.2. Compute atau hitung peubah baru

Pilihan menu **hitung peubah baru** dapat digunakan untuk membentuk variabel baru yang merupakan fungsi dari variabel yang sudah ada. Misalkan saja akan dilakukan transformasi terhadap variabel **tinggi** pada dataset **latihan4** menjadi variabel lain yang dengan nama **tinggi_compute**, yaitu **tinggi:100**.

Langkah-langkah transformasi ini adalah sebagai berikut.

- Pertama-tama, aktifkan dataset yang akan dilakukan transformasi *compute*, yaitu **latihan4** pada **R-Commander**.
- Buka menu **Hitung peubah baru** dengan memilih menu **Data**, kemudian pilih **Atur peubah pada dataset aktif**, dan selanjutnya pilih **Hitung peubah baru**. Selanjutnya akan diperoleh tampilan seperti pada Gambar 2.21.
- Selanjutnya isikan **tinggi_compute** pada kolom **Nama peubah baru** dan **tinggi/100** pada kolom **Eksresi untuk dihitung** seperti yang terlihat pada Gambar 2.21.
- Klik OK, dan sekarang jika dilihat pada dataset **latihan4**, akan diperoleh variabel baru yaitu **tinggi_compute** yang berisikan data hasil transformasi *compute* pada variabel tinggi. Lakukan dengan klik **Lihat dataset**, sehingga diperoleh tampilan data baru pada kolom terakhir yaitu **tinggi_compute** yang merupakan hasil bagi dari variabel **tinggi** dengan **100**.



Gambar 2.21. Jendela pilihan Hitung peubah baru

Pengaturan atau transformasi lain pada dataset yang aktif dapat pula dilakukan dengan menjalankan menu **Data**, pilih **Atur peubah pada dataset aktif**, dan arahkan pada transformasi yang akan diterapkan. Bagian ini hanya menjelaskan dua transformasi awal dari pilihan menu yang ada, yaitu *recode* dan *compute*. Transformasi lain yang dapat dilakukan pada **R-Commander** adalah :

- **Tambahkan** banyaknya pengamatan ke data set
- **Standarisasi** peubah
- **Konversi** peubah **numerik** ke **faktor**
- **Bin** peubah numerik
- **Atur ulang** level faktor
- Definisikan **kontras** dan **faktor**
- **Namakan** ulang peubah
- **Hapus** peubah dari dataset

BAB 3

MANAJEMEN DATA DI R DENGAN COMMAND LINE

Pada **R**, data yang ada dipandang sebagai suatu **objek** yang memiliki suatu **attributes** atau **sifat**. Sifat data ditentukan oleh **type data** dan **mode data**. Ada berbagai **type data** yang dikenal oleh **R**, antara lain **vektor**, **matriks**, **list**, **data frame**, **array**, **factor**, dan **function (built-in command)**. Sedangkan **mode data** yang dikenal **R** ada 4 macam seperti yang terlihat pada Tabel 3.1 berikut ini.

Tabel 3.1. Empat macam **mode data** yang dikenal **R**

Mode	Contoh perintah di Command Line
Numeric	> 23 > c(2.3, 2, 1.3, 3.2) > data.bulan = c(1,2,3,4,5,6,7,8,9,10,11,12)
Complex	> 1+5i > sqrt(as.complex(-5))
Logical	> c(T,F,F,T,T,F,F,T,T,T) > data.tahun > 1998
Character	> c("Budi", "Wati", "Rony", "Naily") > c("F", "T", "2")

Nama **objek** dalam **R** harus dimulai dengan huruf, ditambah dengan kombinasi dari huruf besar, huruf kecil, angka dan titik. Penggunaan titik biasanya dilakukan untuk memudahkan pengorganisasian data. Berikut ini adalah beberapa contoh dari nama objek yang **valid**.

```
databudi
data.budi
data.budi.1
data.budi.5
data.budi.no7.02.02.08
```

Contoh dari nama **objek** yang tidak valid (**invalid**) adalah sebagai berikut:

```
1databudi      : dimulai dari angka
data-budi      : operator - tidak dapat digunakan
databudi=1     : operator = tidak dapat digunakan
```

Dalam **R** versi **2.7.2** ini, **assignment** dapat digunakan dengan operator "**<=**" dan "**=**". Untuk melihat isi dari suatu data **objek**, dapat dilakukan dengan mengetikkan nama objek tersebut di **R prompt** pada **R-console**.

3.1. Jenis-jenis Data Objek

Pada bagian ini akan dijelaskan beberapa jenis data objek pada **R**, yaitu **data array satu dimensi** atau **data vektor**, **data matriks**, **data frame**, dan **data list**.

3.1.1. Data Array Satu Dimensi atau Data Vektor

Vektor merupakan suatu array atau himpunan bilangan, **character** atau **string**, **logical value**, dan merupakan **objek** paling dasar yang dikenal dalam **R**. Pada data vektor harus digunakan mode tunggal pada data, sehingga gabungan dua data atau lebih yang berbeda mode tidak dapat dilakukan kedalam satu objek vektor. Jika ini dilakukan, maka **R** akan mengubah data ke mode yang lebih umum, seperti contoh berikut ini.

```
> c(T,1:10)
[1] 1 1 2 3 4 5 6 7 8 9 10

> c("A",F,T)
[1] "A"  "FALSE" "TRUE"

> c("A",2,4,F,T)
[1] "A"  "2"  "4"  "FALSE" "TRUE"

> x=c(1:10)
> x
[1] 1 2 3 4 5 6 7 8 9 10

> mode(x)
[1] "numeric"

> length(x)
[1] 10
```

Pada contoh pertama dapat dilihat bahwa pada **command line** menghasilkan vektor yang semua data diubah menjadi **mode numerik**, sedangkan pada contoh kedua dan ketiga menghasilkan vektor yang semua datanya diubah menjadi **mode karakter**. Untuk mengetahui **mode** suatu **objek** vektor dapat dilakukan dengan menggunakan **command mode** seperti pada contoh diatas. Jumlah atau panjang data yang bertipe vektor dapat diketahui dengan memanfaatkan fungsi **length** (perhatikan contoh diatas).

Ekstraksi sebagian data vektor dapat dilakukan dengan berbagai cara atau langkah. Dalam praktek analisis data statistik, ekstraksi ini biasanya dilakukan untuk pembentukan data baru berdasarkan data yang sudah ada. Berikut ini adalah beberapa contoh hasil ekstraksi dari suatu data vektor yang terdiri dari 10 elemen, yaitu 10, 5, 14, 12, 8, 11, 9, 10, 16, 20.

```

> x=c(10, 5, 14, 12, 8, 11, 9, 10, 16, 20)
> x  # untuk melihat semua elemen objek vektor x
[1] 10 5 14 12 8 11 9 10 16 20

> x[2]  # menampilkan elemen kedua
[1] 5

> x[c(1,3,7)]  # menampilkan elemen ke-1,3,7
[1] 10 14 9

> x[-c(2,8)]  # menampilkan semua elemen kecuali elemen ke-2,8
[1] 10 14 12 8 11 9 16 20

> x[x>10]  # menampilkan semua elemen yang lebih besar dari 10
[1] 14 12 11 16 20

> y=x[x>10]  # menyimpan vektor yg elemennya lebih besar dari 10 dgn nama y
> y
[1] 14 12 11 16 20

```

3.1.2. Data Matriks

Matriks atau data array dua dimensi adalah salah satu tipe data yang banyak digunakan dalam pemrograman statistik. Sebagian besar fungsi-fungsi statistik dalam **R** dapat dianalisis dengan menggunakan bentuk matriks. Bentuk matriks ini juga banyak digunakan pada operasi fungsi-fungsi **built-in** untuk aljabar linear dalam **R**, seperti untuk penyelesaian suatu persamaan linear.

Proses **entry data** matriks dilakukan dengan menggunakan fungsi **matrix**. Argumen yang diperlukan adalah elemen-elemen dari matriks, dan argumen **optional** yaitu banyaknya baris **nrow** dan banyaknya kolom **ncol**. Sebagai contoh, gunakan perintah-perintah berikut ini pada **R-console**.

```

> matriks.1 = matrix(c(1,2,3,4,5,6),nrow=2,ncol=3)
> matriks.2 = matrix(1:6,nrow=2,ncol=3)
> matriks.3 = matrix(1:6,nrow=2)
> matriks.4 = matrix(1:6,2)
> matriks.1
      [,1] [,2] [,3]
[1,]  1   3   5
[2,]  2   4   6

```

Keempat perintah diatas akan menghasilkan matriks yang sama. Untuk mengetahuinya ketikkan matriks.2, matriks.3, matriks.4, dan kemudian enter untuk masing-masing perintah tersebut.

Pada **R**, data secara **default** akan diisikan kolom perkolom seperti yang terlihat pada contoh berikut ini.

```
> data=c(6.4,8.8,7.5,5.3,7.6,9.5)
> data
[1] 6.4 8.8 7.5 5.3 7.6 9.5

> matriks.a=matrix(data,nrow=3,ncol=2)
> matriks.a
      [,1] [,2]
[1,]  6.4  5.3
[2,]  8.8  7.6
[3,]  7.5  9.5
```

Pengisian matriks menurut baris perbaris dapat dilakukan dengan menggunakan argumen **optional byrow=T** pada **command matrix**. Berikut ini adalah contoh tentang penggunaan argumen tersebut.

```
> matriks.b=matrix(data,nrow=3,ncol=2,byrow=T)
> matriks.b
      [,1] [,2]
[1,]  6.4  8.8
[2,]  7.5  5.3
[3,]  7.6  9.5

> dim(matriks.a)
[1] 3 2

> length(matriks.a)
[1] 6

> mode(matriks.a)
[1] "numeric"
```

Dimensi, **length** dan **mode** dari suatu matriks dapat dilihat dengan menggunakan perintah **dim**, **length**, dan **mode** seperti pada contoh diatas. Perlu diingat bahwa semua elemen dari matriks harus memiliki **mode** yang sama. Jika hal ini tidak dipenuhi, maka elemen-elemen akan diubah menjadi **mode** yang paling umum.

Ada beberapa operator yang biasa digunakan untuk operasi matriks dan vektor, antara lain perkalian, invers matriks, **transpose** matriks dan **crossproduct**. Ringkasan dari operator-operator ini dapat dilihat pada Tabel 2.2.

Tabel 3.2. Operator untuk operasi matriks dan vektor

Operator	Keterangan
*	Perkalian elemen demi elemen dari matriks
%*%	Perkalian matriks
%O%	Outer
solve	Invers dari suatu matriks
t	Transpose dari suatu matriks
crossprod	Crossproduct suatu matriks, yaitu $t(x) \%* \% x$

Berikut ini adalah beberapa contoh hasil penggunaan operator pada suatu matriks dan vektor.

```

> a=1:5
> a
[1] 1 2 3 4 5
> a*a # perkalian elemen demi elemen dari matriks a
[1] 1 4 9 16 25
> crossprod(a) # crossproduct dari matriks a, yaitu t(a) %* % a
[1]
[1,] 55
> b=matrix(c(1:4),2)
> b
[1,] [2,]
[1,] 1 3
[2,] 2 4
> b*b # perkalian elemen demi elemen dari matriks b
[1,] [2,]
[1,] 1 9
[2,] 4 16
> b%* %b # perkalian matriks b dengan matriks b
[1,] [2,]
[1,] 7 15
[2,] 10 22
> solve(b) # invers dari matriks b
[1,] [2,]
[1,] -2 1.5
[2,] 1 -0.5

```

Pada **R**, dapat pula dilakukan penggabungan satu kolom atau satu baris baru kedalam matriks lain. Hal ini dapat dilakukan dengan menggunakan perintah **rbind** (untuk menambahkan ke baris) dan **cbind** (untuk menambahkan ke kolom). Perhatikan contoh-contoh berikut ini.

```
> a=matrix(c(3,4,5,6,7,8),2,3)
> a
      [,1] [,2] [,3]
[1,]  3   5   7
[2,]  4   6   8

> a1=cbind(a,c(1,2)) # menambahkan ke kolom ke-4 dari a
> a1
      [,1] [,2] [,3] [,4]
[1,]  3   5   7   1
[2,]  4   6   8   2

> a2=cbind(c(1,2),a) # menambahkan ke kolom ke-1 dari a
> a2
      [,1] [,2] [,3] [,4]
[1,]  1   3   5   7
[2,]  2   4   6   8

> a3=rbind(a,c(1,2,3)) # menambahkan ke baris ke-3 dari a
> a3
      [,1] [,2] [,3]
[1,]  3   5   7
[2,]  4   6   8
[3,]  1   2   3

> a4=rbind(c(1,2,3),a) # menambahkan ke baris ke-1 dari a
> a4
      [,1] [,2] [,3]
[1,]  1   2   3
[2,]  3   5   7
[3,]  4   6   8
```

3.1.3. Data Frame

Data frame merupakan objek yang mempunyai bentuk sama dengan matriks, yaitu terdiri atas baris dan kolom. Perbedaannya adalah **data frame** dapat terdiri atas **mode** data yang berbeda-beda untuk setiap kolomnya. Misalkan saja, kolom pertama adalah **numeric**, kolom kedua adalah **string/character**, dan kolom ketiga adalah **logical**. Objek **data frame** dapat dibuat dengan menggunakan perintah **data.frame**, seperti pada contoh-contoh berikut ini.

```
> data.frame(c(1:4),c(T,T,F,F))
  c.1.4. c.T..T..F..F.
1  1    TRUE
2  2    TRUE
3  3   FALSE
4  4   FALSE

> data.frame(nomer=c(1:4),jawaban=c(T,T,F,F)) # ada nama kolom
  nomer jawaban
1  1    TRUE
2  2    TRUE
3  3   FALSE
4  4   FALSE

> cobaframe=data.frame(c(1:4),c(T,T,F,F)) # simpan objek di cobaframe
> cobaframe
  c.1.4. c.T..T..F..F.
1  1    TRUE
2  2    TRUE
3  3   FALSE
4  4   FALSE

> names(cobaframe)[1]="nomer" # nama kolom ke-1 "nomer"
> names(cobaframe)[2]="jawaban" # nama kolom ke-2 "jawaban"
> cobaframe
  nomer jawaban
1  1    TRUE
2  2    TRUE
3  3   FALSE
4  4   FALSE

> cobaframe1=data.frame(c(1:4),c(T,T,F,F))
> cobaframe1
  c.1.4. c.T..T..F..F.
1  1    TRUE
2  2    TRUE
3  3   FALSE
4  4   FALSE

> names(cobaframe1)=c("nomer","jawaban") # beri nama kolom
> cobaframe1
  nomer jawaban
1  1    TRUE
2  2    TRUE
3  3   FALSE
4  4   FALSE
```

Secara umum, perintah-perintah diatas adalah ekuivalen dengan perintah berikut ini.

```
> cobaframe2=data.frame(nomer=c(1:4),jawaban=c(T,T,F,F))
> cobaframe2
  nomer jawaban
1    1   TRUE
2    2   TRUE
3    3  FALSE
4    4  FALSE
```

Seperti pada data **vektor**, ekstraksi sebagian data pada **matriks** dan **data frame** dapat pula dilakukan dengan berbagai cara atau langkah. Berikut ini adalah beberapa contoh hasil ekstraksi dari suatu **matriks** dan **data frame**.

```
> matriks.1=matrix(1:9,3)
> dataframe.1=data.frame(nomer=1:4,nama=c("Adi","Budi","Cika","Dony"),
                        nilai=7:10)

> matriks.1
     [,1] [,2] [,3]
[1,]  1  4  7
[2,]  2  5  8
[3,]  3  6  9

> matriks.1[2,2]
[1] 5

> dataframe.1
  nomer nama nilai
1    1  Adi    7
2    2  Budi    8
3    3  Cika    9
4    4  Dony   10

> dataframe.1[2,2]
[1] Budi
Levels: Adi Budi Cika Dony

> dataframe.1["nama"]
  nama
1  Adi
2  Budi
3  Cika
4  Dony
```

3.1.4. Data List

Data list merupakan objek yang paling umum atau general dan paling fleksibel di dalam **R**. List adalah suatu vektor terurut dari sekumpulan komponen. Setiap komponen dapat berupa sembarang data objek, yaitu **vektor**, **matriks**, **data frame**, atau **data list** sendiri. Tiap komponen pada **data list** dapat mempunyai **mode** yang berbeda. **Data list** dapat dibuat dengan menggunakan perintah **list**. Berikut ini adalah contoh pendefinisian dan pemakaian elemen **list**.

```
> list(c(1:3),c(T,F,T,T),data.frame(nama=c("Budi","Cika","Dony"),nilai=c(8:10)))
[[1]]
[1] 1 2 3

[[2]]
[1] TRUE FALSE TRUE TRUE

[[3]]
  nama nilai
1 Budi    8
2 Cika    9
3 Dony   10

> datalist.1=list(nomer=c(1:3),jawaban=c(T,F,T,T),nilaiframe=data.frame(nama
=c("Budi","Cika","Dony"),nilai=c(8:10)))
> datalist.1
$nomer
[1] 1 2 3

$jawaban
[1] TRUE FALSE TRUE TRUE

$nilaiframe
  nama nilai
1 Budi    8
2 Cika    9
3 Dony   10
```

Seperti pada jenis-jenis data sebelumnya, ekstraksi sebagian data pada **data list** dapat pula dilakukan dengan berbagai cara atau langkah. Berikut ini adalah beberapa contoh hasil ekstraksi dari suatu **data list**.

```

> datalist.1[1]    # mengakses nama dan elemen pertama
$nomer
[1] 1 2 3

> datalist.1[[1]]  # mengakses elemen pertama
[1] 1 2 3

> datalist.1$nomer  # mengakses elemen pertama berdasarkan namanya
[1] 1 2 3

> datalist.1$jawaban # mengakses elemen kedua
[1] TRUE FALSE TRUE TRUE

> datalist.1$nilaiframe # mengakses elemen dataframe
  nama nilai
1 Budi   8
2 Cika   9
3 Dony  10

> datalist.1$nilaiframe$nama
[1] Budi Cika Dony
Levels: Budi Cika Dony

```

3.2. Importing Data pada Command Line

Secara umum, proses **importing** data pada **R** dapat dilakukan dengan dua cara, yaitu menggunakan perintah-perintah di **command line** dan menggunakan fasilitas **GUI R-Cmdr** (lihat bagian 2.1 sebelumnya). Pada bagian ini akan dijelaskan penggunaan perintah pada **command line** untuk **importing** data.

3.2.1. Membaca File ASCII

Suatu file **ASCII** biasanya terdiri dari bilangan-bilangan yang dipisahkan menggunakan spasi, tab, tanda akhir baris atau tanda baris baru, serta pembatas yang lain. Misalkan data file **ASCII** yang dibuat di **NOTEPAD** dengan nama **latihan5.txt** berisi data seperti berikut ini.

50	28	75	35	49	64	88	94	54	34	28	56
87	42	33	67	31	98	58	47	37	66	64	25
66	35	87	58	93	86	69	29	96	86	57	80

Anggap bahwa file **ASCII** dengan nama **latihan5.txt** ini sudah tersimpan pada direktori kerja **R**. Proses impor data dapat dilakukan dengan perintah **scan** dan **latihan5.txt** sebagai argumennya. Apabila data tidak berada pada direktori kerja **R**, maka tulis juga direktori tersebut pada argumennya. Berikut ini adalah contoh proses impor data file **ASCII**.

```
> scan("latihan5.txt")
Read 36 items
[1] 50 28 75 35 49 64 88 94 54 34 28 56 87 42 33 67 31 98 58 47 37 66 64 25 66
[26] 35 87 58 93 86 69 29 96 86 57 80

> data5.scan=scan("latihan5.txt")
Read 36 items

> data5.scan
[1] 50 28 75 35 49 64 88 94 54 34 28 56 87 42 33 67 31 98 58 47 37 66 64 25 66
[26] 35 87 58 93 86 69 29 96 86 57 80

> matrix5.scan=matrix(scan("latihan5.txt"),6)
Read 36 items

> matrix5.scan
  [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 50 88 87 58 66 69
[2,] 28 94 42 47 35 29
[3,] 75 54 33 37 87 96
[4,] 35 34 67 66 58 86
[5,] 49 28 31 64 93 57
[6,] 64 56 98 25 86 80

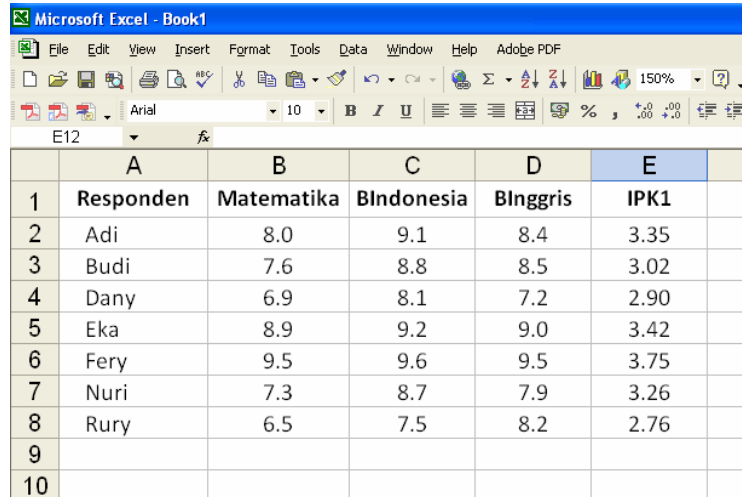
> data6.scan=scan("c:\\Kerja_R\\latihan5.txt")
Read 36 items

> data6.scan
[1] 50 28 75 35 49 64 88 94 54 34 28 56 87 42 33 67 31 98 58 47 37 66 64 25 66
[26] 35 87 58 93 86 69 29 96 86 57 80
```

3.2.2. Importing Data File Excel

Data file **Excel** dengan ekstensi **.XLS** dapat diimpor secara langsung menggunakan fasilitas **GUI R-Cmdr** (lihat bagian sebelumnya). Untuk dapat diimpor ke dalam **R** dengan fasilitas command line, maka data file **Excel** harus terlebih dulu diubah menjadi format **Text Tab Delimited** (ekstensi **.TXT**) atau **CSV comma delimited** (ekstensi **.CSV**). Setelah itu, data ini dapat diimpor menggunakan perintah **read.table** atau **read.csv**.

Misalkan saja data file **Excel** yang akan diimpor adalah seperti pada gambar berikut ini dan telah disimpan menjadi file **data1.txt** atau **data1.csv**.



	A	B	C	D	E
1	Responden	Matematika	BIndonesia	BInggris	IPK1
2	Adi	8.0	9.1	8.4	3.35
3	Budi	7.6	8.8	8.5	3.02
4	Dany	6.9	8.1	7.2	2.90
5	Eka	8.9	9.2	9.0	3.42
6	Fery	9.5	9.6	9.5	3.75
7	Nuri	7.3	8.7	7.9	3.26
8	Rury	6.5	7.5	8.2	2.76
9					
10					

Gambar 3.1. Jendela **data1.txt** pada **Excel** yang akan diimpor ke **R**

Proses impor **data1.txt** dapat dilakukan dengan perintah **read.table**, sedangkan, impor **data1.csv** dilakukan dengan perintah **read.csv**. Argumen optional **header=T** digunakan dengan tujuan agar **R** menggunakan baris pertama dari file sebagai header atau nama dari variabel. Seperti pada bagian sebelumnya, apabila data tidak berada pada direktori kerja **R**, maka tulis juga direktori tersebut pada argumennya. Berikut ini adalah contoh proses impor data file **dengan ekstensi .TXT dan .CSV**.

```
> latihan2 <- read.table("data1.txt", header=TRUE)           # atau
> latihan2 <- read.table("c:\\Kerja_R\\data1.txt", header=TRUE) # atau
> latihan2 <- read.table("c:/Kerja_R/data1.txt", header=TRUE)

> latihan2
```

	Responden	Matematika	BIndonesia	BInggris	IPK1
1	Adi	8.0	9.1	8.4	3.35
2	Budi	7.6	8.8	8.5	3.02
3	Dany	6.9	8.1	7.2	2.90
4	Eka	8.9	9.2	9.0	3.42
5	Fery	9.5	9.6	9.5	3.75
6	Nuri	7.3	8.7	7.9	3.26
7	Rury	6.5	7.5	8.2	2.76


```
> latihan3 <- read.csv("data1.csv", header=TRUE)

> latihan3
      Responden.Matematika.BIndonesia.BInggris.IPK1
1      Adi;8;9.1;8.4;3.35
2      Budi;7.6;8.8;8.5;3.02
3      Dany;6.9;8.1;7.2;2.9
4      Eka;8.9;9.2;9;3.42
5      Fery;9.5;9.6;9.5;3.75
6      Nuri;7.3;8.7;7.9;3.26
7      Rury;6.5;7.5;8.2;2.76
```

3.2.3. Importing Data dari Paket Statistik

R mempunyai paket atau **library** **foreign** untuk melakukan importing data dari file dalam format paket statistika yang lain. Sampai saat ini yang tersedia pada **R** adalah importing data file dari paket-paket statistika berikut :

- **MINITAB** : gunakan perintah **read.mtp** untuk membaca file '**Minitab Portable Worksheet**' atau data dengan ekstensi **.MTP**. File ini dapat dibuat di **MINITAB** dengan perintah **SAVE AS** dan pilihan **.MTP**
- **SPSS** : gunakan perintah **read.spss** untuk membaca file '**.SAV**'.
- **SAS** : gunakan perintah **read.ssd** atau **read.xport**.
- **S+** : gunakan perintah **read.S**
- **STATA** : gunakan perintah **read.dta**
- **Systat** : gunakan perintah **read.systat**
- **Epi info** : gunakan perintah **read.epiinfo** untuk membaca file '**.REC**'.

Pada bagian ini akan diberikan contoh hanya untuk mengimpor data file **SPSS** dan **MINITAB** yang seringkali digunakan dalam analisis data statistik. Misalkan data file **SPSS** yang sudah dimiliki diberi nama **WORLD95.SAV** dan telah disimpan di direktori kerja **R**. Proses impor data ini ke dalam **R** dengan menggunakan perintah command line adalah sebagai berikut.

```
> latihan4 <- read.spss("World95.sav", use.value.labels=TRUE,
  max.value.labels=Inf, to.data.frame=TRUE)

> latihan4[,1:5] # hanya menampilkan 5 kolom pertama saja
```

	COUNTRY	POPULATN	DENSITY	URBAN	RELIGION
1	Afghanistan	20500	25.0	18	Muslim
2	Argentina	33900	12.0	86	Catholic
3	Armenia	3700	126.0	68	Orthodox
4	Australia	17800	2.3	85	Protstnt
5	Austria	8000	94.0	58	Catholic
6	Azerbaijan	7400	86.0	54	Muslim
...					

Perintah **use.value.labels=TRUE** digunakan untuk mendapatkan variabel yang bertipe **FACTOR** dengan “value label” seperti yang ada pada data file di **SPSS**.

Berikut ini adalah proses impor data file **MINITAB** dalam ekstensi **.MTP** ke dalam **R** dengan menggunakan perintah command line. Misalkan data file **MINITAB** yang sudah dimiliki adalah **FA.MTW** dan telah disimpan ke dalam ekstensi **.MTP** menjadi **FA.MTP**.

```
> latihan5 <- read.mtp("C:/Kerja_R/Fa.MTP")

> latihan5
$X
[1] 10 8 13 9 11 14 6 4 12 7 5

$Y1
[1] 8.04 6.95 7.58 8.81 8.33 9.96 7.24 4.26 10.84 4.82 5.68

$Y2
[1] 9.14 8.14 8.74 8.77 9.26 8.10 6.13 3.10 9.13 7.26 4.74

$Y3
[1] 7.46 6.77 12.74 7.11 7.81 8.84 6.08 5.39 8.15 6.42 5.73

$X4
[1] 8 8 8 8 8 8 19 8 8 8

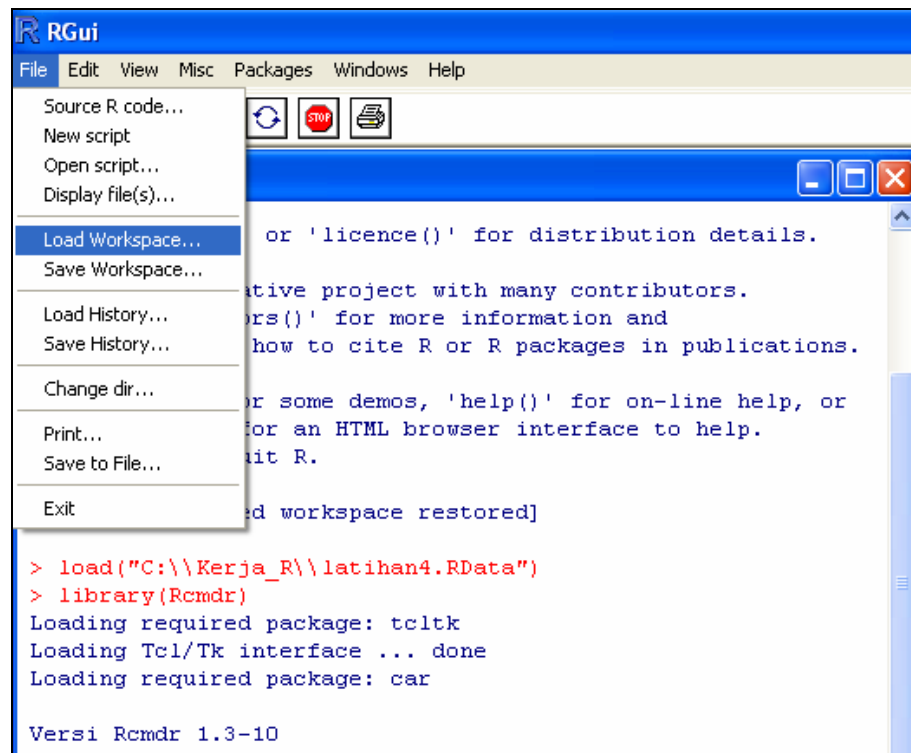
$Y4
[1] 6.58 5.76 7.71 8.84 8.47 7.04 5.25 12.50 5.56 7.91 6.89
```

BAB 4

GRAFIK MENGGUNAKAN R-Commander

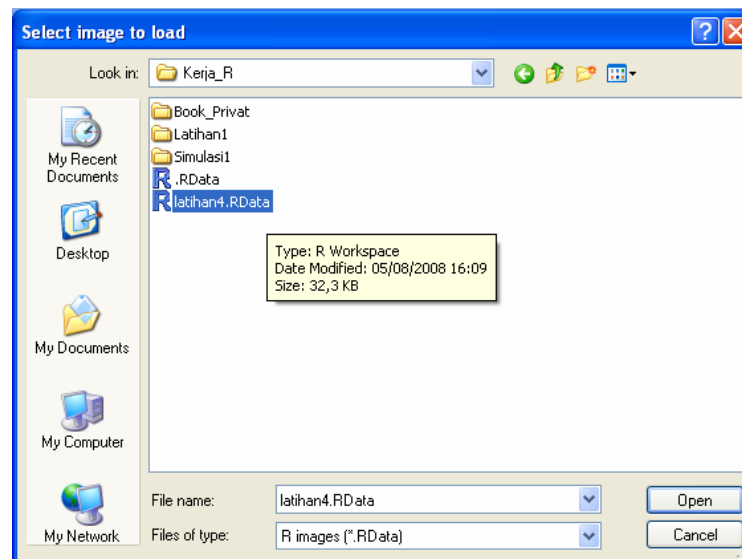
Pada bab ini akan dibahas penggunaan **R-Commander** untuk membuat penyajian statistik deskriptif dari suatu kumpulan data. Fokus utama adalah pembuatan beberapa macam bentuk grafik yang banyak digunakan dalam analisis data.

Sebagai langkah awal, buka kembali program **R** dengan mengklik icon **R 2.7.2**. Kemudian, ubah direktori dimana file workspace berada. Misalkan file **latihan4.RData** (hasil impor data **SPSS** dengan nama file **WORLD95.SAV**) ada di '**C:\Kerja_R**', maka direktori diubah ke **C:\Kerja_R**. Load file workspace tersebut dengan menggunakan menu **File**, pilih **Load Workspace...** seperti pada gambar berikut ini.



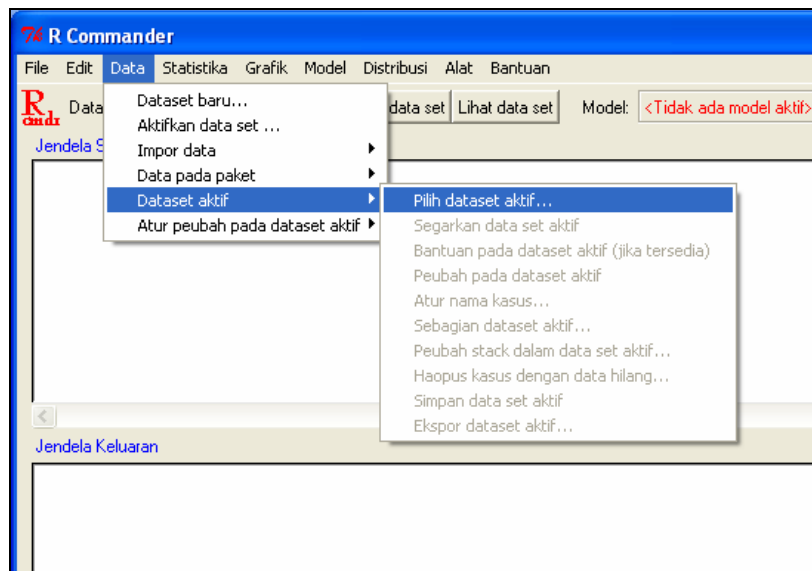
Gambar 4.1. Jendela dialog untuk **Load Workspace**

Setelah diklik **Load Workspace...** maka jendela **R** akan memberikan pilihan direktori dan file workspace mana yang akan ditampilkan, seperti yang terlihat pada Gambar 4.2. Pilihlah file workspace **latihan4.RData** yang ada di direktori **C:\Kerja_R**.



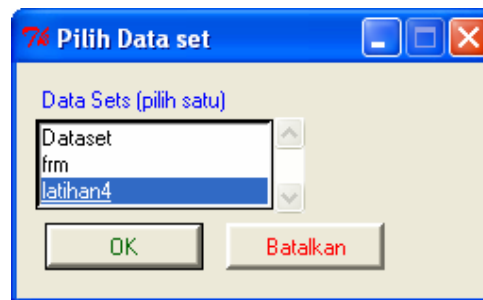
Gambar 4.2. Jendela dialog untuk pilihan **file workspace** yang akan diaktifkan

Langkah selanjutnya adalah mengaktifkan **R-commander** dengan menggunakan perintah **library(Rcmdr)**. Setelah itu, aktifkan dataset dengan menggunakan menu **Data**, klik **Dataset aktif**, dan **Pilih dataset aktif...** seperti yang ditampilkan pada Gambar 4.3.



Gambar 4.3. Jendela dialog untuk memilih **dataset** yang akan diaktifkan

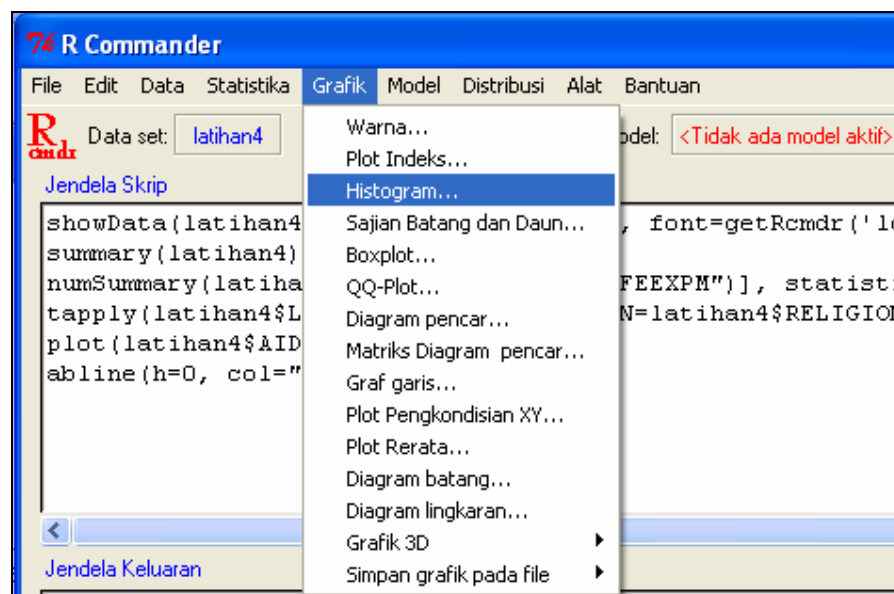
Dari beberapa pilihan **Datasets** yang ada, klik **latihan4** sebagai file workspace yang akan diaktifkan, seperti pada Gambar 4.4. Dengan demikian, proses pengaktifan kembali data latihan4 sudah dilakukan, dan proses analisis data baik secara statistik deskriptif atau inferens dapat dilakukan.



Gambar 4.4. Jendela dialog untuk pilihan **dataset** yang akan diaktifkan

4.1. Grafik dalam R-GUI

R menyediakan banyak menu pilihan grafik pada **R-Commander**, antara lain **Histogram**, **Diagram Batang dan Daun**, **Boxplot**, dan lain-lain. Secara lengkap pilihan grafik yang tersedia dapat dilihat pada gambar berikut ini.



Gambar 4.5. Jendela dialog untuk pilihan **Grafik** pada **R-Commander**

4.2. Grafik Histogram

Menu yang digunakan untuk membuat grafik histogram adalah **Grafik**, pilih **Histogram...**. Misalkan akan dibuat histogram untuk variabel **LIFEEXPF** (usia harapan hidup wanita di suatu negara), maka pada jendela dialog yang muncul, pilih **LIFEEXPF** seperti pada Gambar 4.5. Isikan jumlah interval yang diinginkan pada kolom **Banyaknya bin**, dan klik **OK** untuk menampilkan output histogramnya.

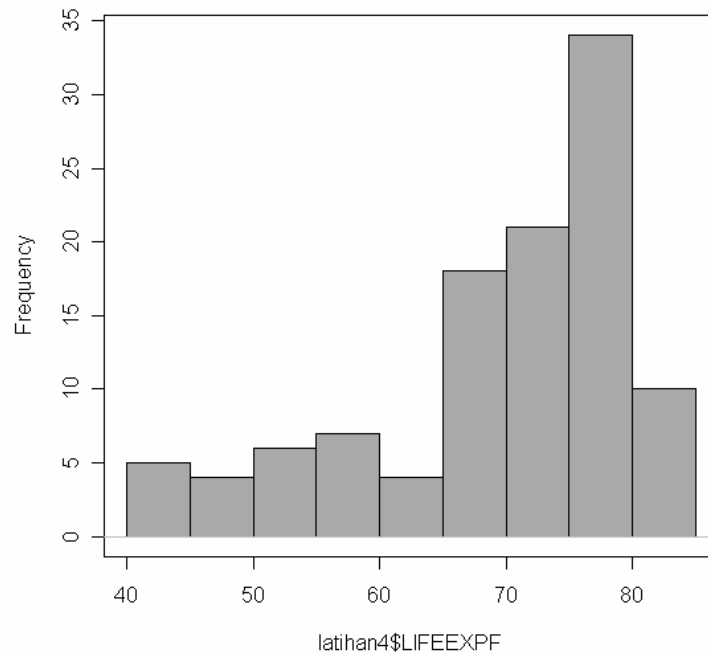


Gambar 4.5. Jendela dialog pilihan variabel untuk pembuatan histogram

Output histogram untuk data **LIFEEXPF** yang diperoleh dari perintah di atas dapat dilihat pada Gambar 4.6. Dalam contoh ini, digunakan metode **auto** untuk pemilihan jumlah interval, yaitu metode **Sturges** dan **Cacahan Frekuensi** yang digunakan untuk nilai (**Skala Sumbu**) yang diplotkan pada histogram. Selain itu dapat digunakan pilihan **Persentase** atau **Kepadatan** pada **Skala Sumbu**.

Output histogram ini dapat disimpan dengan menggunakan menu **File**, dan pilih **Save as** dari jendela grafik. Pilihlah output yang sesuai, misalkan saja dalam format **PDF**. Maka pilih format PDF dalam daftar format file output. Selanjutnya, beri nama file output dengan **histogramLIFEEXPF.PDF**. Selain itu, output histogram ini dapat pula disimpan dalam format **Metafile**, **Postcript**, **Png**, **Bmp**, dan **Jpeg**.

Jika file histogram ini ingin dikopi untuk di insert kedalam program lain, misalkan kedalam **Microsoft Word**, maka dapat digunakan menu **File**, pilih **Copy to the clipboard**, dan pilih **as a Bitmap** atau **Ctrl-C**. Kemudian, buka program **Microsoft Word**, maka file grafik dapat di **paste** kan menggunakan perintah **Ctrl-V**.



Gambar 4.6. Output histogram pada variabel **LIFEXPF**

Selain menggunakan menu di **R-Commander**, pembuatan histogram dapat juga dilakukan dengan command line di **R-Console**, yaitu dengan command **hist** diikuti argumen optional yang diinginkan. Berikut adalah contoh pembuatan histogram dengan command line untuk variabel **LIFEXPF** dan **LIFEXPM** (usia harapan hidup pria di suatu negara).

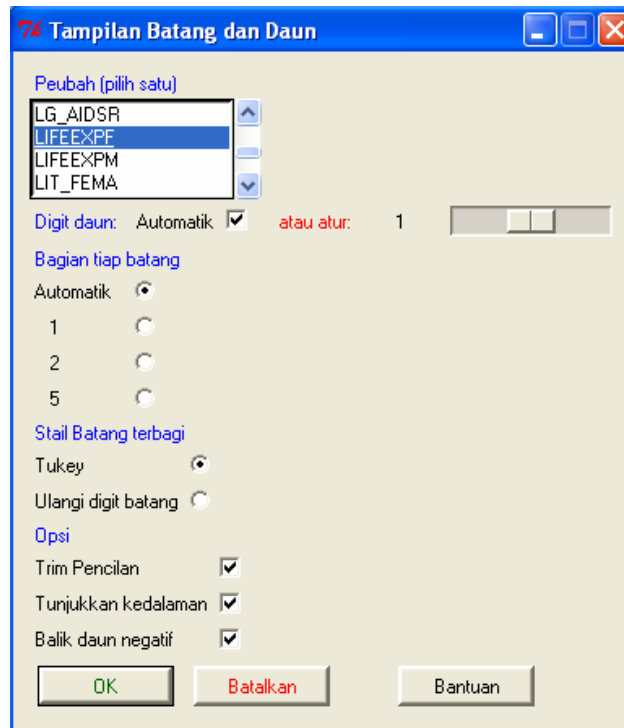
```
> Hist(latihan4$LIFEXPF, scale="frequency", breaks="Sturges", col="darkgray")
> Hist(latihan4$LIFEXPF, scale="frequency", breaks=10, col="darkgray")
> hist(latihan4$LIFEXPF)

> # lihat perbedaan output histogram yang dihasilkan

> Hist(latihan4$LIFEXPM, scale="frequency", breaks="Sturges", col="darkgray")
> Hist(latihan4$LIFEXPM, scale="frequency", breaks=10, col="darkgray")
> hist(latihan4$LIFEXPM)
```

4.3. Diagram Batang dan Daun (Stem-and-Leaf)

Menu yang digunakan untuk membuat diagram batang dan daun adalah **Grafik**, pilih **Sajian Batang dan Daun...**. Misalkan akan dibuat diagram batang dan daun untuk variabel **LIFEEXPF**, maka pada jendela dialog yang muncul, pilih **LIFEEXPF** seperti pada Gambar 4.7.



Gambar 4.7. Jendela dialog untuk pembuatan **diagram batang dan daun**

Isikan argumen optional yang diinginkan pada kolom-kolom yang tersedia, dan klik **OK** untuk menampilkan output diagram batang dan daun. Output dari diagram ini akan ditampilkan di **Jendela Keluaran** pada **R-Commander** seperti pada Gambar 4.8.

Output tersebut menjelaskan bahwa bilangan pada daun menunjukkan nilai-nilai satuan. Sehingga dapat diinterpretasikan bahwa usia harapan hidup wanita yang terendah adalah 43 tahun dan yang tertinggi adalah 82 tahun. Ada 3 (tiga) negara dengan usia harapan hidup wanitanya sebesar 82 tahun. Dalam contoh ini, pilihan **Automatik** menghasilkan diagram batang dan daun dengan jumlah kelas dalam setiap batang adalah 5 kelas interval.


```

> stem.leaf(latihan4$LIFEEXPF)

1 | 2: represents 12
  leaf unit: 1
    n: 109

LO: 43 44 44 45 45 46 47
    9   5* | 00
    12   t | 223
    15   f | 455
    17   s | 77
    22   5. | 88889
        6* |
    23   t | 3
    26   f | 455
    32   s | 677777
    39   6. | 8888899
    45   7* | 000001
    51   t | 222333
  (14)   f | 44444555555555
    44   s | 66666777777888888888
    24   7. | 9999999
    17   8* | 00000001111111
     3   t | 222

```

Gambar 4.8. Output diagram batang dan daun pada variabel **LIFEEXPF**

Pembuatan diagram batang dan daun ini dapat juga dilakukan dengan command line di **R-Console**, yaitu dengan command **stem.leaf** diikuti argumen optional yang diinginkan. Berikut adalah contoh pembuatan diagram batang dan daun dengan command line untuk variabel **LIFEEXPF** dan **LIFEEXPM**.

```

> stem.leaf(latihan4$LIFEEXPF)
> stem.leaf(latihan4$LIFEEXPF, m=2)
> stem.leaf(latihan4$LIFEEXPF, style="bare", unit=1)

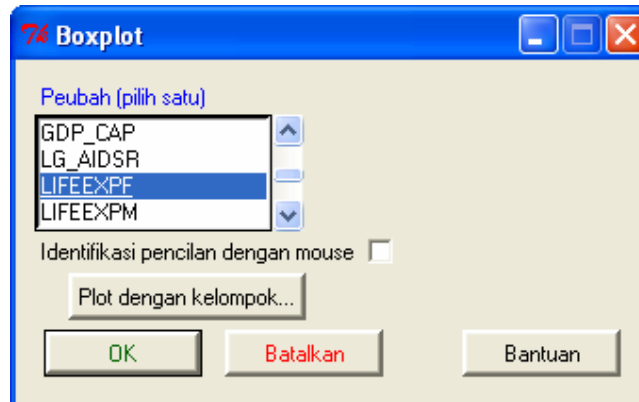
> # lihat perbedaan output diagram batang dan daun yang dihasilkan

> stem.leaf(latihan4$LIFEEXPM)
> stem.leaf(latihan4$LIFEEXPM, m=3)
> stem.leaf(latihan4$LIFEEXPF, style="bare", unit=1)

```

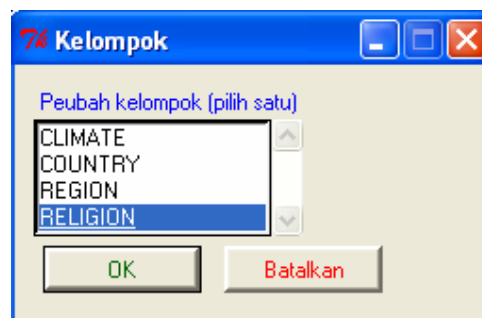
4.4. Grafik BoxPlot

R menyediakan pilihan **Boxplot...** pada menu **Grafik** untuk membuat tampilan **BoxPlot** dari suatu data. Misalkan akan dibuat **BoxPlot** untuk variabel **LIFEEXPF** berdasarkan **RELIGION** (kelompok agama mayoritas di negara tersebut), maka pada jendela dialog yang muncul, pilih **LIFEEXPF** seperti pada Gambar 4.9.



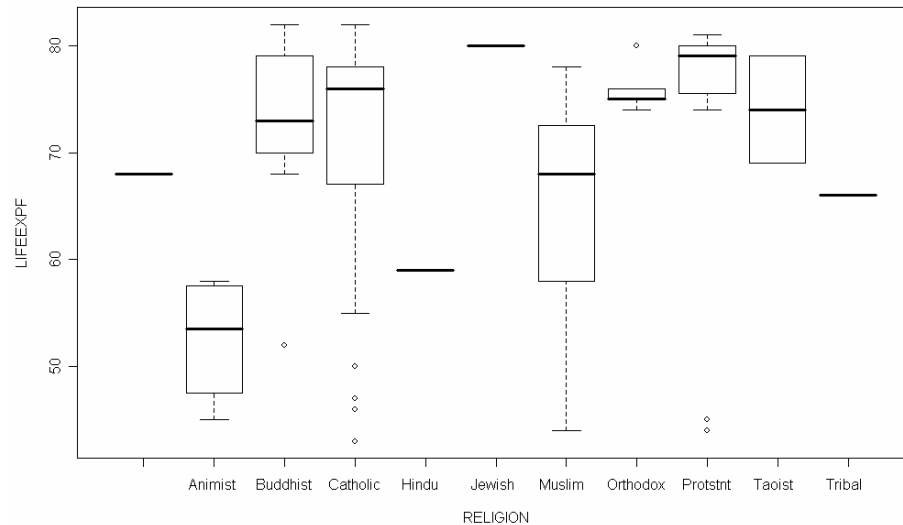
Gambar 4.9. Jendela dialog untuk pilihan variabel dalam pembuatan **Boxplot**

Setelah itu, pilih **Plot dengan kelompok...** sehingga diperoleh tampilan jendela seperti pada Gambar 4.10. Klik **RELIGION** sebagai variabel kelompok, dan kemudian klik **OK**.



Gambar 4.10. Jendela dialog untuk pilihan variabel kelompok dalam **Boxplot**

Output dari **BoxPlot** yang diperoleh akan ditampilkan di **Jendela Keluaran** pada **R-Commander** seperti pada Gambar 4.11. Output tersebut menjelaskan bahwa usia harapan hidup wanita di negara dengan mayoritas penduduknya beragama Jewish (Yahudi) dan Protestan secara rata-rata adalah paling tinggi dibanding lainnya.



Gambar 4.11. Output **BoxPlot** pada variabel **LIFEEXPF** berdasarkan **RELIGION**

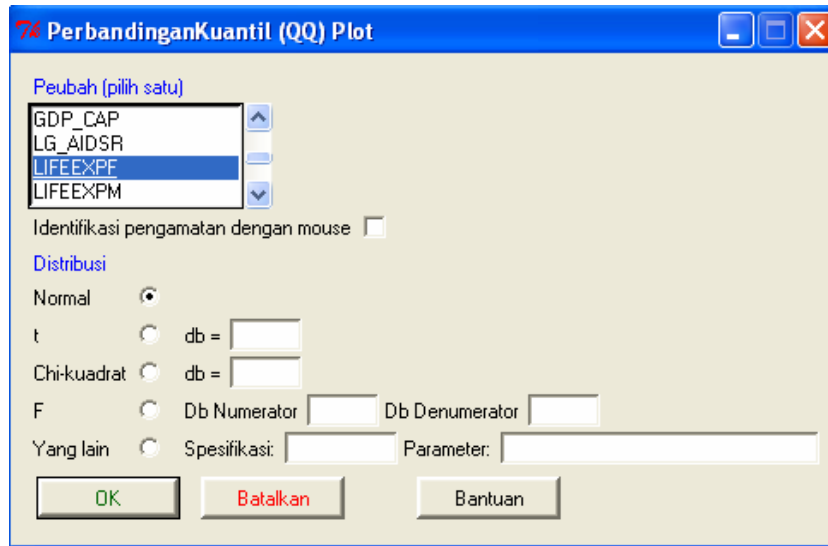
Command line di **R-Console** dapat juga digunakan untuk pembuatan **BoxPlot**, yaitu dengan command **boxplot** diikuti argumen optional yang diinginkan. Berikut adalah contoh pembuatan **BoxPlot** dengan command line untuk variabel **LIFEEXPF** dan **LIFEEXPM** sendiri-sendiri dan berdasarkan variabel **RELIGION**.

```
> boxplot(latihan4$LIFEEXPF)
> boxplot(latihan4$LIFEEXPM)
> boxplot(LIFEEXPF~RELIGION, ylab="LIFEEXPF", xlab="RELIGION", data=latihan4)
> boxplot(latihan4$LIFEEXPF~latihan4$RELIGION)

> # lihat perbedaan output Box-Plot yang dihasilkan
```

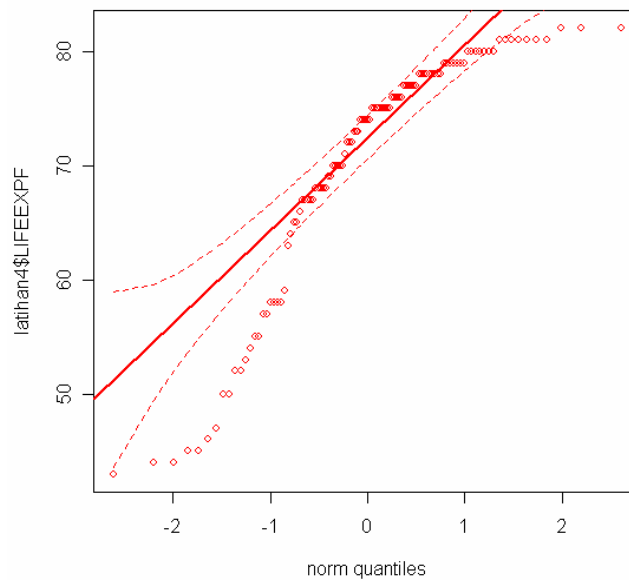
4.5. Grafik QQ-Plot

QQ-Plot merupakan salah satu metode eksplorasi secara grafik yang dapat digunakan untuk menguji apakah suatu data berdistribusi normal. Untuk membuat grafik **QQ-Plot**, **R** menyediakan pilihan **QQ-Plot...** pada menu **Grafik**. Misalkan akan dibuat **QQ-Plot** untuk variabel **LIFEEXPF**, maka pada jendela dialog yang muncul, pilih **LIFEEXPF** seperti pada Gambar 4.12.



Gambar 4.12. Jendela dialog untuk pilihan variabel dalam pembuatan **QQ-Plot**

Kemudian pilih **LIFEEXPF** dari daftar variabel dan gunakan distribusi normal sebagai distribusi default pada **QQ-Plot**. Klik **OK**, maka akan diperoleh grafik seperti berikut.



Gambar 4.13. Output **QQ-Plot** pada variabel **LIFEEXPF**

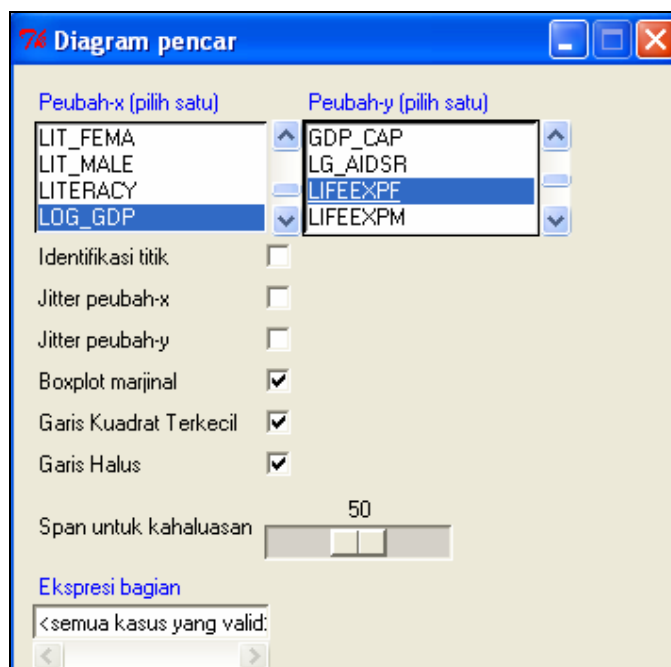
Berdasarkan output pada Gambar 4.13 dapat dijelaskan bahwa variabel **LIFEEXPF** tidak berdistribusi normal dan data cenderung menceng ke kanan (ekor lebih panjang di bagian kiri). Hal ini terlihat jelas juga dari grafik histogramnya (lihat Gambar 4.6).

Command line di **R-Console** dapat juga digunakan untuk pembuatan QQ-Plot, yaitu dengan command **boxplot** diikuti argumen optional yang diinginkan. Berikut adalah contoh pembuatan **BoxPlot** dengan command line untuk variabel **LIFEEXPF** dan **LIFEEXPM** sendiri-sendiri dan berdasarkan variabel **RELIGION**.

```
> qq.plot(latihan4$LIFEEXPF, dist= "norm", labels=FALSE)
> qq.plot(latihan4$LIFEEXPM, dist= "norm", labels=FALSE)
```

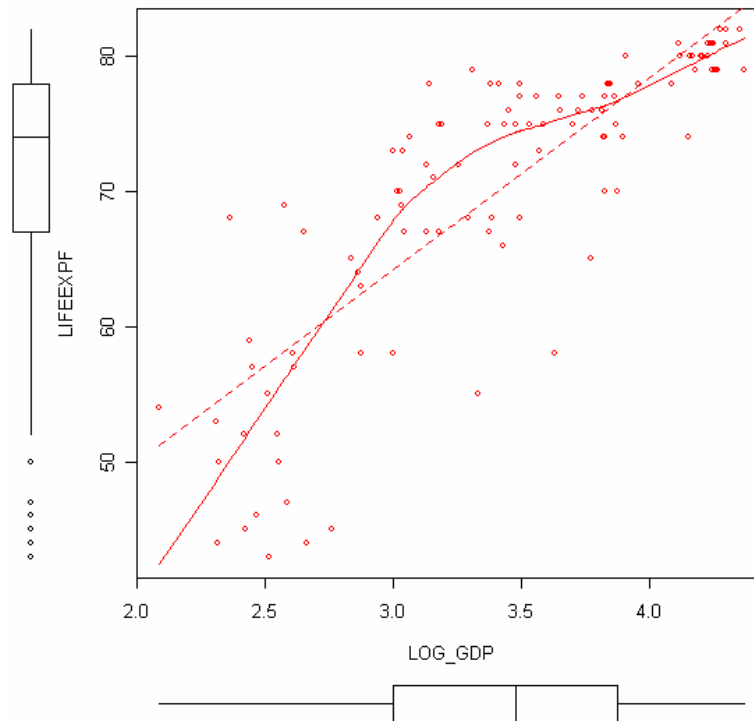
4.6. Grafik Diagram Pencar (ScatterPlot)

R menyediakan pilihan **Diagram pencar...** pada menu **Grafik** untuk membuat tampilan **ScatterPlot** dari suatu data. Misalkan akan dibuat **ScatterPlot** untuk variabel **LIFEEXPF** sebagai sumbu Y dan variabel **LOGGDP** sebagai sumbu X. Gunakan **default** untuk pilihan yang lain, seperti pada Gambar 4.14.



Gambar 4.14. Jendela dialog pilihan variabel dalam pembuatan **Diagram Pencar**

Kemudian pilih **LOG_GDP** pada variabel X dan **LIFEEXPF** untuk variabel Y, dan klik **OK** sehingga diperoleh output grafik seperti berikut ini.



Gambar 4.15. Output **Diagram Pencar** antara variabel **LOG_GDP** dan **LIFEEXPF**

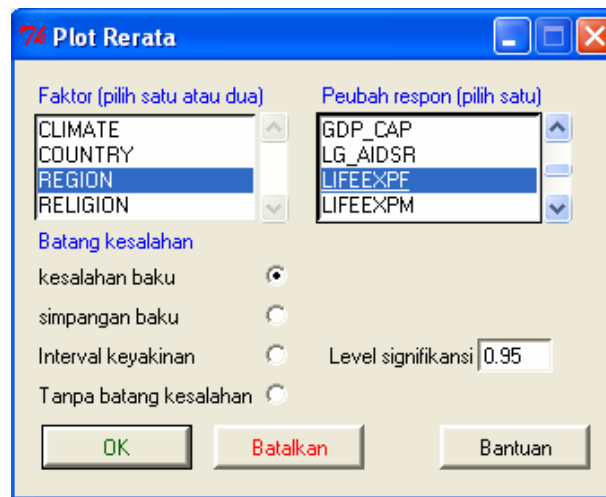
Pada output **Diagram Pencar**, diperoleh juga grafik **Box-Plot** dari setiap marginal variabel, dan garis regresi linear dan non-parametrik terbaik untuk menggambarkan hubungan antara kedua variabel ini.

Command line di **R-Console** dapat juga digunakan untuk pembuatan **Diagram Pencar** di atas, yaitu dengan command **scatterplot** diikuti argumen optional yang diinginkan. Berikut adalah contoh pembuatan **Diagram Pencar** dengan command line untuk variabel **LIFEEXPF** sebagai sumbu Y, dan **LOG_GDP** sebagai sumbu X, seperti perintah di **R-Commander** di atas.

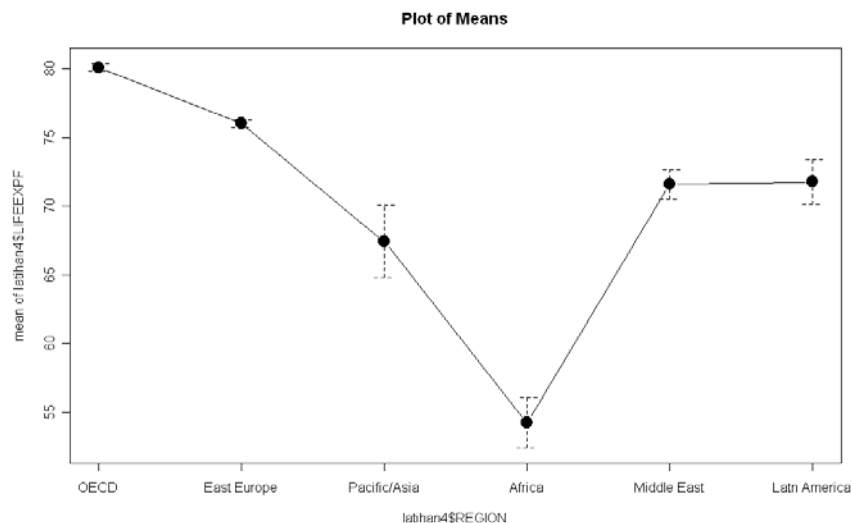
```
> scatterplot(LIFEEXPF~LOG_GDP, reg.line=lm, smooth=TRUE, labels=FALSE,
              boxplots='xy', span=0.5, data=latihan4)
```

4.7. Grafik Plot Rata-rata (Mean)

R menyediakan pilihan **Plot Rerata...** pada menu **Grafik** untuk membuat tampilan **Plot Rata-rata** dari suatu data. Misalkan akan dibuat Plot Rata-rata untuk variabel **LIFEEXPF** berdasarkan **REGION** (kelompok wilayah negara), maka pada jendela dialog yang muncul, pilih **REGION** dan **LIFEEXPF** seperti pada Gambar 4.16. Klik **OK**, sehingga diperoleh output seperti pada Gambar 4.17.



Gambar 4.16. Jendela dialog pilihan variabel dalam pembuatan **Plot Rata-rata**



Gambar 4.17. Output **Plot Rata-rata** variabel **REGION** dan **LIFEEXPF**

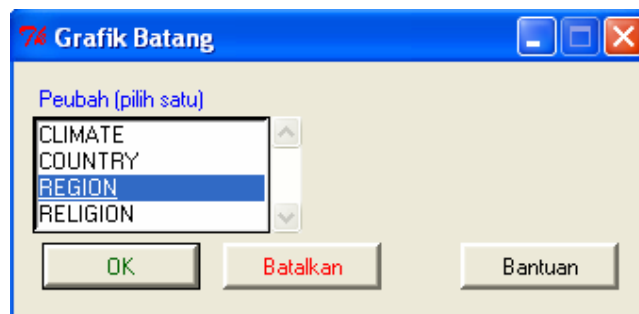
Pada output Plot Rata-rata di atas dapat dilihat bahwa **LIFEEXPF** (usia harapan hidup wanita) yang terendah rata-ratanya adalah pada negara-negara di Afrika.

Command line di **R-Console** untuk pembuatan **Plot Rata-rata** adalah command **plotMeans** diikuti argumen optional yang diinginkan. Berikut adalah contoh pembuatan **Plot Rata-rata** dengan command line untuk variabel **LIFEEXPF** dan **REGION**.

```
> plotMeans(latihan4$LIFEEXPF, latihan4$REGION, error.bars="se")  
> plotMeans(latihan4$LIFEEXPF, latihan4$REGION, error.bars="conf.int", level=0.95)
```

4.8. Diagram Batang (Bar-Chart)

R menyediakan pilihan **Diagram batang...** pada menu **Grafik** untuk membuat tampilan **Diagram Batang** dari suatu data. Misalkan akan dibuat **Diagram Batang** untuk variabel **REGION**, maka pada jendela dialog yang muncul, pilih **REGION** seperti pada Gambar 4.18 berikut ini.

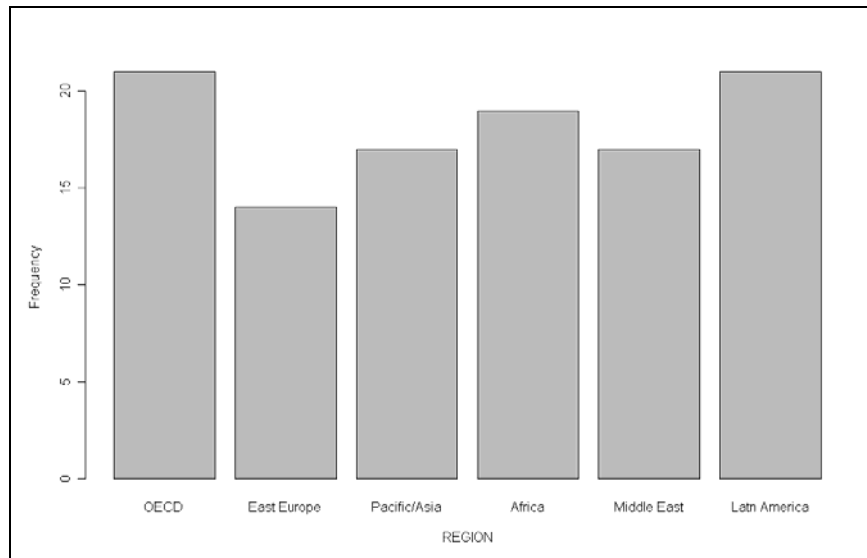


Gambar 4.18. Jendela dialog pilihan variabel dalam pembuatan **Diagram Batang**

Setelah itu klik **OK**, dan akan diperoleh output **Diagram Batang** seperti pada Gambar 4.19. Pada output tersebut dapat dilihat bahwa ada dua kelompok **REGION** terbesar, yaitu negara-negara yang termasuk di regional OECD dan Amerika Latin.

Command line di **R-Console** dapat juga digunakan untuk pembuatan **Diagram Batang**, yaitu dengan command **barplot** diikuti argumen optional yang diinginkan. Berikut adalah contoh pembuatan **Diagram Batang** dengan command line untuk variabel **REGION**, seperti perintah di **R-Commander** di atas.

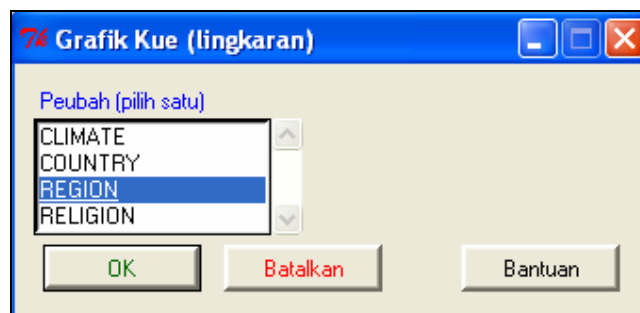
```
> barplot(table(latihan4$REGION), xlab="REGION", ylab="Frequency")
```

Gambar 4.19. Output **Diagram Batang** dari variabel **REGION**

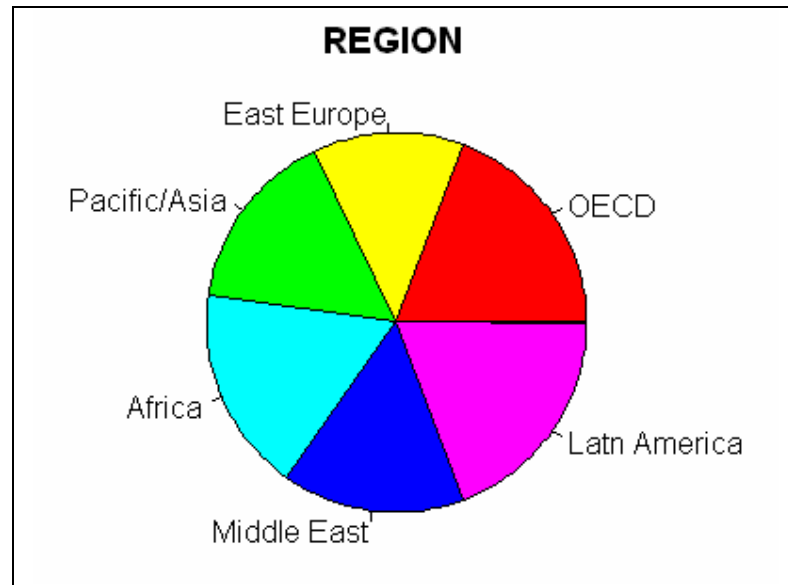
4.9. Diagram Lingkaran (Pie-Chart)

Tampilan **Diagram Lingkaran** pada paket **R** disediakan melalui pilihan **Diagram lingkaran...** pada menu **Grafik**. Misalkan akan dibuat Diagram Lingkaran untuk variabel **REGION**, maka pada jendela dialog yang muncul, pilih **REGION** seperti pada Gambar 4.20 berikut ini.



Gambar 4.20. Jendela dialog pilihan variabel dalam pembuatan Diagram Lingkaran

Kemudian klik **OK**, dan akan diperoleh output **Diagram Lingkaran** seperti yang terlihat pada Gambar 4.21.



Gambar 4.21. Output **Diagram Lingkaran** dari variabel **REGION**

Command line di **R-Console** yang dapat digunakan untuk pembuatan **Diagram Lingkaran** adalah **pie** diikuti argumen optional yang diinginkan. Berikut adalah contoh pembuatan **Diagram Lingkaran** dengan command line untuk variabel **REGION**, seperti perintah di **R-Commander** di atas.

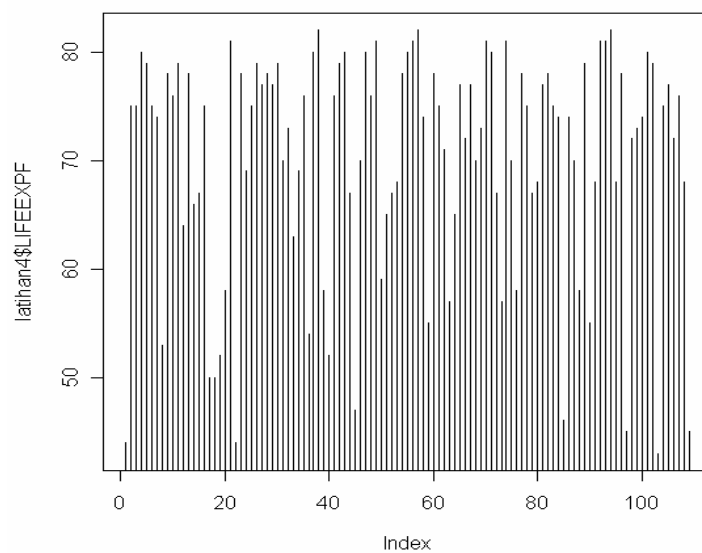
```
> pie(table(latihan4$REGION), labels=levels(latihan4$REGION), main="REGION",
      col=rainbow(length(levels(latihan4$REGION))))
```

4.10. Plot Indeks

Plot Indeks adalah suatu plot dari variabel menurut indeks atau urutan data. Plot ini dalam analisis data statistik lebih dikenal dengan **Time Series Plot**. **R** menyediakan pilihan **Plot Indeks...** pada menu **Grafik** untuk membuat tampilan **Plot Indeks** dari suatu data. Pada **R-Commander** ini hanya tersedia dua pilihan tipe dari plot, yaitu **Paku** dan **Poin**. Misalkan akan dibuat **Plot Indeks** untuk variabel **LIFEEXPF**, maka pada jendela dialog yang muncul, pilih **LIFEEXPF** seperti pada Gambar 4.22. Dalam hal ini, pilih tipe plot **Paku**, dan kemudian klik **OK**, sehingga diperoleh output **Plot Indeks** seperti yang terlihat pada Gambar 4.23.



Gambar 4.22. Jendela dialog pilihan variabel dalam pembuatan **Plot Indeks**

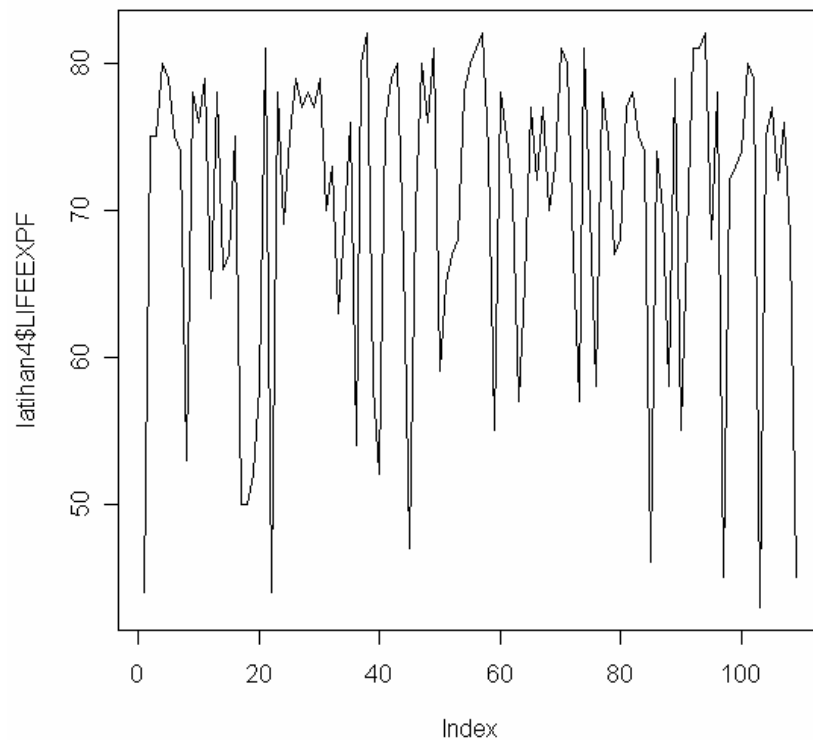


Gambar 4.23. Output **Plot Indeks** dari variabel **LIFEEXPF**

Command line di **R-Console** yang dapat digunakan untuk pembuatan **Plot Indeks** adalah **plot** diikuti argumen optional yang diinginkan. Jika akan menampilkan plot berupa garis, maka dapat digunakan pilihan **type="l"**, yang berarti **line** atau garis. Berikut adalah contoh pembuatan **Plot Indeks** dengan command line untuk variabel **LIFEEXPF**.

```
> plot(latihan4$LIFEEXPF, type="h")  
> plot(latihan4$LIFEEXPF, type="p")  
> plot(latihan4$LIFEEXPF, type="l", main="Time Series Plot Data LIFEEXPF")
```

Berikut ini adalah output **Plot Indeks** pada variabel **LIFEEXPF** dengan pilihan tipe garis (**line**) yang dinotasikan dengan "l".



Gambar 4.24. Output **Plot Indeks** dari variabel **LIFEEXPF** dengan **type="l"**

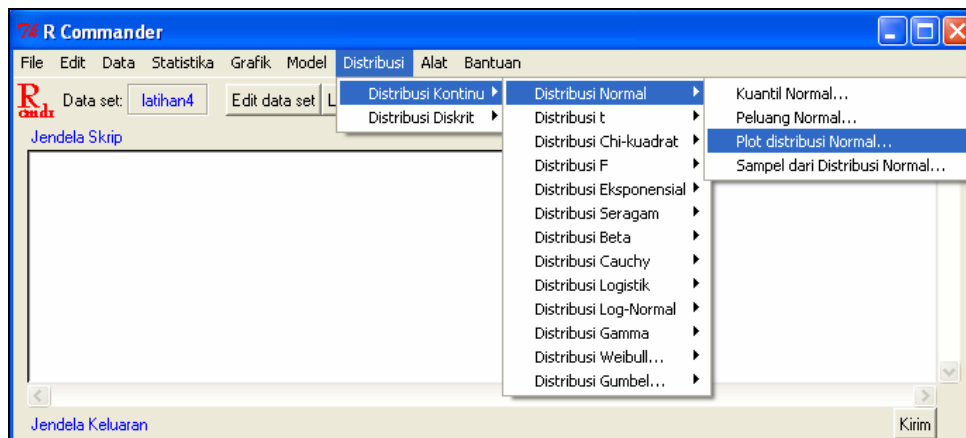
BAB 5

FUNGSI DISTRIBUSI PELUANG DI R-Commander

Pada bab ini akan dijelaskan penggunaan **R-Commander** untuk perhitungan yang berkaitan dengan fungsi distribusi peluang. **R-Commander** menyediakan menu untuk melakukan beberapa operasi standar yang berkaitan dengan fungsi distribusi peluang, yaitu :

- Perhitungan nilai **kuantil**
- Perhitungan nilai **peluang**
- Pembuatan **plot distribusi** atau **grafik densitas**
- Pembuatan **plot distribusi kumulatif**
- **Pembangkitan data** atau **random data**

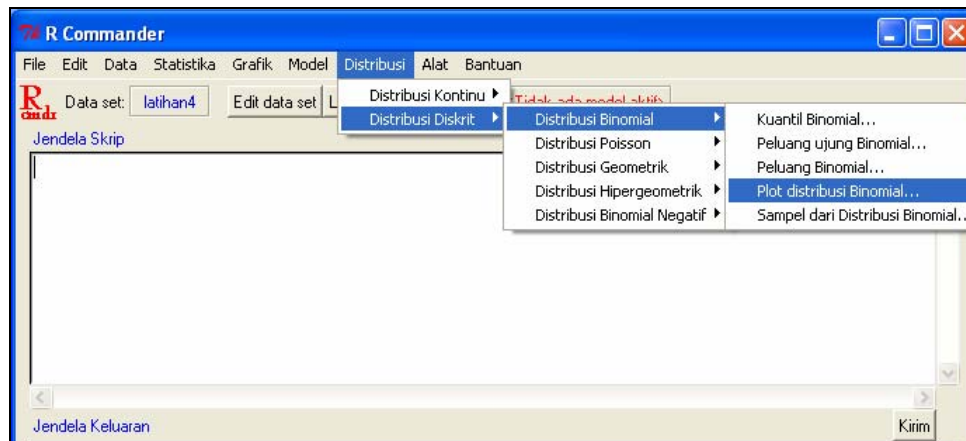
Secara umum ada dua macam distribusi yang disediakan paket **R**, yaitu **Distribusi Kontinu** dan **Diskrit**. Untuk mengetahui distribusi kontinu atau diskrit apa saja yang ada di **R**, dapat dilakukan dengan memilih menu **Distribusi**, kemudian pilih **Distribusi Kontinu**, sehingga akan muncul pilihan dari berbagai distribusi kontinu yang ada di **R**, seperti yang terlihat pada Gambar 5.1.



Gambar 5.1. Jendela dialog untuk pilihan **Distribusi Kontinu**

Dari Gambar 5.1 dapat dilihat macam-macam distribusi kontinu yang ada di **R**, yaitu Distribusi **Normal**, **t**, **Chi-kuadrat**, **F**, **Ekspensial**, **Seragam**, **Beta**, **Cauchy**, **Logistik**, **Log-Normal**, **Gamma**, **Weibull**, dan **Gumbel**. Secara umum, proses perhitungan yang berkaitan dengan distribusi peluang untuk macam-macam distribusi kontinu tersebut adalah relatif sama. Untuk itu, pada bab ini fokus pembahasan hanya diberikan pada distribusi yang banyak dipakai di analisis statistika dasar, yaitu Distribusi Normal.

Distribusi Diskrit yang disediakan di **R** dapat dilihat dengan memilih menu **Distribusi**, kemudian pilih **Distribusi Diskrit**, sehingga akan muncul pilihan dari berbagai distribusi diskrit yang ada di **R**, seperti yang terlihat pada Gambar 5.2. Dari gambar ini dapat dilihat bahwa distribusi diskrit yang ada di **R** adalah Distribusi **Binomial**, **Poisson**, **Geometrik**, **Hipergeometrik**, dan **Binomial Negatif**.



Gambar 5.2. Jendela dialog untuk pilihan **Distribusi Diskrit**

5.1. Fungsi Distribusi Kontinu

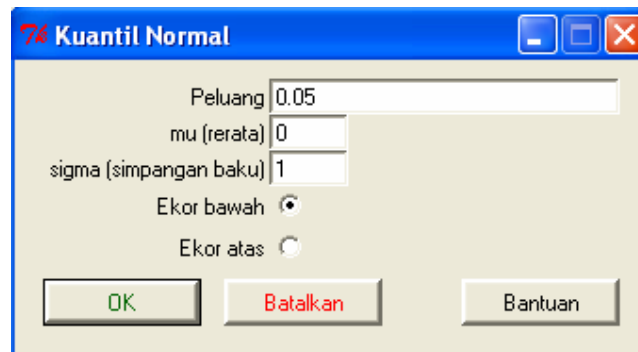
Pada bagian ini akan dijelaskan cara perhitungan berkaitan dengan fungsi distribusi peluang, yaitu perhitungan nilai kuantil, pembuatan plot atau grafik densitas, pembuatan plot distribusi kumulatif, dan pembangkitan data dari distribusi kontinu, khususnya Distribusi Normal yang banyak digunakan dalam analisis statistika dasar. Secara umum, fungsi kepadatan probabilitas dari Distribusi Normal adalah sebagai berikut

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \text{ untuk } -\infty < x < \infty$$

dengan parameter μ adalah nilai rata-rata, dan σ adalah deviasi standar.

5.1.1. Menghitung Kuantil dari Distribusi Normal

Perhitungan nilai kuantil tertentu dari Distribusi Normal dapat dilakukan dengan **R-Commander**, yaitu gunakan menu **Distribusi**, pilih **Distribusi Kontinu**, pilih **Distribusi Normal**, dan kemudian klik **Kuantil Normal...**. Setelah itu akan terlihat jendela pilihan untuk mendapatkan kuantil yang akan dicari seperti pada Gambar 5.3.



Gambar 5.3. Jendela dialog untuk perhitungan **Kuantil Normal**

Misalkan akan dihitung nilai kuantil $\alpha=0,05$ (5%) dari **Distribusi Normal Standar**, yaitu ingin dicari nilai Z_{α} sedemikian hingga

$$P(Z \leq Z_{\alpha}) = 0,05 \quad (\text{luasan } \textit{lower tail} \text{ atau ekor bawah}),$$

maka pada jendela isian **Peluang** tulis nilai 0.05. Dalam hal ini rata-rata adalah 0 dan deviasi standar 1. Kemudian klik **OK**, sehingga akan diperoleh nilai pada jendela keluaran **R-Commander** yaitu $Z_{0,05} = -1.644854$. Pilihan ekor atas atau *upper tail* digunakan jika ingin dicari nilai $Z_{1-\alpha}$ sedemikian hingga

$$P(Z \leq Z_{1-\alpha}) = 1 - \alpha \quad (\text{luasan } \textit{upper tail} \text{ atau ekor atas}).$$

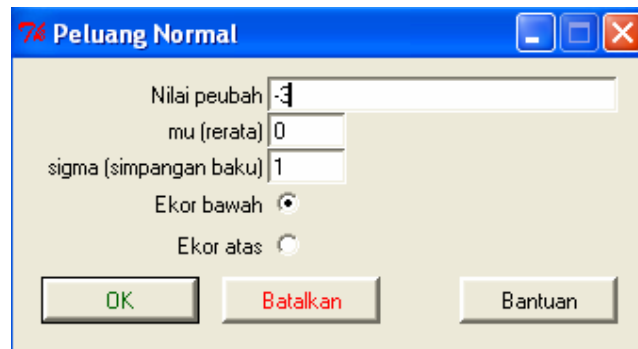
Jika pilihan ekor atas yang digunakan, maka keluaran **R-Commander** memberikan nilai 1.644854 pada jendela keluarannya.

Selain menggunakan menu di **R-Commander**, perhitungan kuantil normal dapat juga dilakukan dengan command line di **R-Console**, yaitu dengan command **qnorm** diikuti argumen optional yang diinginkan. Berikut adalah contoh perhitungan kuantil normal dengan command line untuk $\alpha=0,05$.

```
> qnorm(c(0.05), mean=0, sd=1, lower.tail=TRUE)
[1] -1.644854
> qnorm(c(0.05), mean=10, sd=2, lower.tail=TRUE)
[1] 6.710293
> qnorm(c(0.05), mean=0, sd=1, lower.tail=FALSE)
[1] 1.644854
> qnorm(c(0.05), mean=10, sd=2, lower.tail=FALSE)
[1] 13.28971
```

5.1.2. Menghitung Peluang dari Distribusi Normal

Perhitungan peluang dari suatu nilai tertentu dari Distribusi Normal dapat dilakukan dengan **R-Commander**, yaitu gunakan menu **Distribusi**, pilih **Distribusi Kontinu**, pilih **Distribusi Normal**, dan kemudian klik **Peluang Normal...**. Setelah itu akan terlihat jendela pilihan untuk memperoleh peluang yang dicari seperti pada Gambar 5.4 berikut ini.



Gambar 5.4. Jendela dialog untuk perhitungan **Peluang Normal**

Ada empat isian utama dari jendela dialog untuk perhitungan **Peluang Normal**, yaitu **Nilai peubah**, **mu**, **sigma**, dan pilihan **Ekor bawah** atau **Ekor atas**. Secara matematis, fasilitas ini dapat digunakan untuk menghitung

$$P(X \leq c) = \dots ? \quad (\text{luasan } \textit{lower tail} \text{ atau ekor bawah}),$$

dan

$$P(X \geq c) = \dots ? \quad (\text{luasan } \textit{upper tail} \text{ atau ekor atas}),$$

dari suatu peubah (variabel) random X yang berdistribusi Normal, atau $X \sim N(\mu, \sigma)$.

Misalkan akan dihitung nilai peluang dari **Distribusi Normal Standar**, yaitu ingin dicari nilai

$$P(Z \leq -3) = \dots ? \quad (\text{luasan } \textit{lower tail} \text{ atau ekor bawah}),$$

maka pada jendela isian **Nilai peubah** tulis nilai **-3**. Dalam hal ini rata-rata adalah 0 dan deviasi standar 1. Klik **OK**, sehingga akan diperoleh nilai 0.001349898 pada jendela keluaran **R-Commander**. Pilihan ekor atas atau *upper tail* digunakan jika ingin dicari nilai

$$P(Z \geq c) = \dots ? \quad (\text{luasan } \textit{upper tail} \text{ atau ekor atas}).$$

Jika pilihan ekor atas yang digunakan dan $c = 3$, maka keluaran **R-Commander** juga akan memberikan nilai 0.001349898 pada jendela keluarannya.

Perhitungan peluang normal dapat juga dilakukan dengan command line di **R-Console**, yaitu dengan command **pnorm** diikuti argumen optional yang diinginkan. Berikut adalah contoh perhitungan peluang normal dengan command line untuk berbagai nilai peubah.

```
> pnorm(c(-3), mean=0, sd=1, lower.tail=TRUE)
[1] 0.001349898

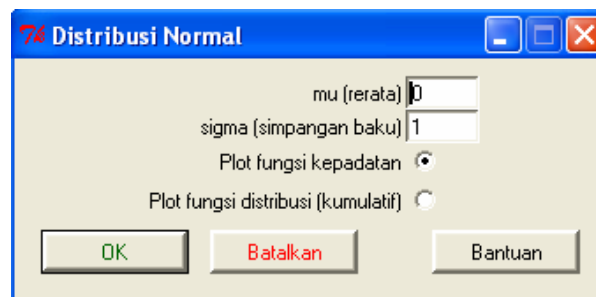
> pnorm(c(6.710293), mean=10, sd=2, lower.tail=TRUE)
[1] 0.05000001

> pnorm(c(3), mean=0, sd=1, lower.tail=FALSE)
[1] 0.001349898

> pnorm(c(13.28971), mean=10, sd=2, lower.tail=FALSE)
[1] 0.049999986
```

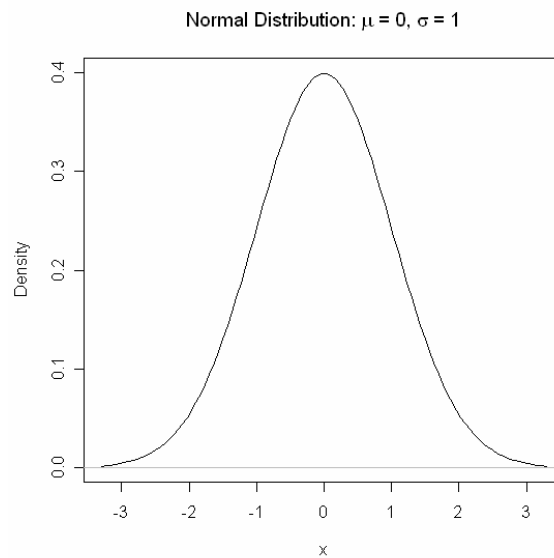
5.1.3. Membuat Plot dari Distribusi Normal

Plot dari Distribusi Normal teoritis dengan rata-rata dan deviasi standar tertentu dapat dilakukan dengan **R-Commander**, yaitu gunakan menu **Distribusi**, pilih **Distribusi Kontinu**, pilih **Distribusi Normal**, dan kemudian klik **Plot Distribusi Normal...**. Setelah itu akan terlihat jendela pilihan untuk mendapatkan plot distribusi normal teoritis yang ingin dicari seperti pada Gambar 5.5 di bawah ini.



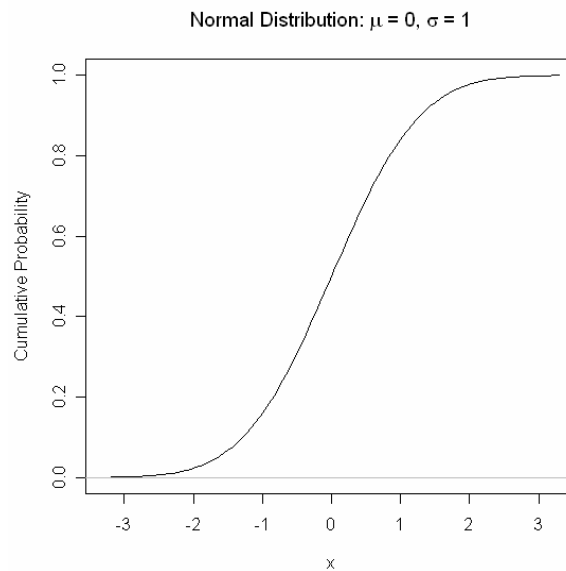
Gambar 5.5. Jendela dialog untuk pembuatan **Plot Distribusi Normal**

Misalkan akan dibuat plot fungsi kepadatan peluang dari **Distribusi Normal Standar**, maka pada jendela isian **mu (rerata)** tulis nilai **0** dan **sigma (simpangan baku)** **1**. Klik pilihan **Plot fungsi kepadatan**, dan kemudian klik **OK**, sehingga akan diperoleh plot fungsi kepadatan dari Distribusi Normal Standar seperti pada Gambar 5.6 berikut ini.



Gambar 5.6. Output plot fungsi kepadatan **Distribusi Normal Standar**

Jika pilihan **Plot fungsi distribusi (kumulatif)** yang dipilih, maka akan diperoleh output plot fungsi distribusi kumulatif dari Distribusi Normal Standar seperti terlihat pada Gambar 5.7.



Gambar 5.7. Output plot fungsi distribusi kumulatif dari **Distribusi Normal Standar**

Pembuatan plot fungsi kepadatan dan fungsi distribusi kumulatif dapat juga dilakukan dengan command line di **R-Console**, yaitu dengan command **dnorm** (untuk plot fungsi kepadatan) dan command **pnorm** (untuk plot fungsi distribusi kumulatif) diikuti argumen optional yang diinginkan. Berikut adalah contoh pembuatan plot fungsi kepadatan dengan command line untuk suatu nilai peubah.

```
> .x <- seq(-3.291, 3.291, length=100)
> plot(.x, dnorm(.x, mean=0, sd=1), xlab="x", ylab="Density",
      main=expression(paste("Normal Distribution: ", mu, " = 0, ",
      sigma, " = 1")), type="l")
> abline(h=0, col="gray")
```

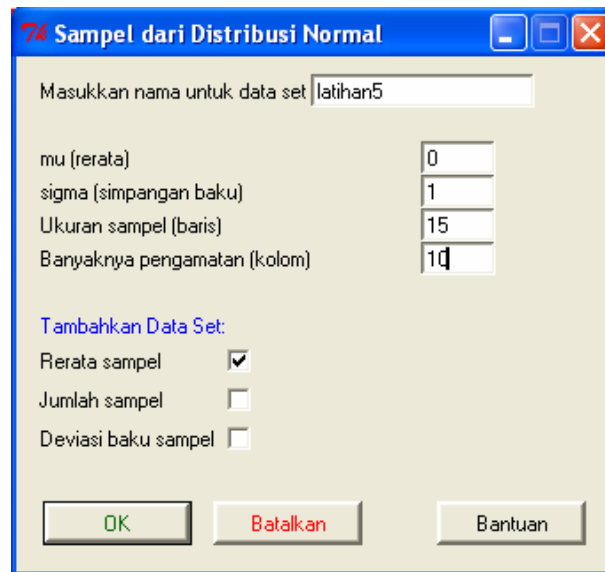
Sedangkan contoh pembuatan plot fungsi distribusi kumulatif dengan command line untuk suatu nilai peubah adalah seperti berikut.

```
> .x <- seq(-4, 4, length=100)
> plot(.x, pnorm(.x, mean=0, sd=1), xlab="x", ylab="Cumulative Probability",
      main=expression(paste("Normal Distribution: ", mu, " = 0, ",
      sigma, " = 1")), type="l")
> abline(h=0, col="gray")
> # perhatikan perbedaan output yang dihasilkan
```

5.1.4. Membangkitkan Data dari Distribusi Normal

R menyediakan fasilitas untuk membangkitkan data yang mengikuti distribusi statistika tertentu. Misalkan akan dibangkitkan data yang mengikuti distribusi normal, maka dapat digunakan menu **Distribusi**, pilih **Distribusi Kontinu**, pilih **Distribusi Normal**, dan kemudian klik **Sampel dari Distribusi Normal...**. Setelah itu akan terlihat jendela pilihan untuk pembangkitan data dari distribusi normal seperti pada Gambar 5.8.

Sebagai contoh, akan dibangkitkan data sebanyak 15 baris dan 10 kolom yang mengikuti **Distribusi Normal Standar**, maka tulis nama dataset hasil dari data bangkitan pada isian **Masukkan nama untuk data set** (misalkan dengan nama **latihan5**). Pada jendela isian **mu (rerata)** tulis nilai **0**, **sigma (simpangan baku)** **1**, **Ukuran sampel (baris)** **15**, dan **Banyaknya pengamatan (kolom)** **10**. Jika rata-rata sampel juga ingin ditambahkan, maka klik pada pilihan **Rerata sampel**, dan kemudian klik **OK**. Pilihan-pilihan yang lain, yaitu **Jumlah sampel** dan **Deviasi baku sampel** juga dapat ditampilkan jika diinginkan.



Gambar 5.8. Jendela dialog untuk membangkitkan data dari **Distribusi Normal**

Untuk mengetahui hasil data yang dibangkitkan, klik pilihan **Lihat data set** pada **R-Commander**, sehingga akan terlihat data-data hasil bangkitan seperti pada Gambar 5.9. Secara umum akan diperoleh 15 baris sampel dan 11 kolom data, yaitu 10 kolom data hasil bangkitan dan 1 kolom terakhir yang berisi rata-rata dari setiap sampel yang dibangkitkan.

latihan5					
	obs1	obs2	obs3	obs4	obs5
sample1	-2.39499806	0.78128054	-1.15472307	0.08485773	-0.631691902
sample2	1.00791838	-0.06138112	0.84571234	-0.94164965	0.319022398
sample3	0.15369008	-0.72340084	-0.73274082	0.49822236	0.731495030
sample4	0.74635667	1.15390883	0.84342999	0.28261438	0.003457076
sample5	-1.03967804	0.68299094	0.02112067	0.44256125	1.478325947
sample6	-0.03362661	-1.03782377	0.33254214	-0.22858938	1.607264850
sample7	-0.45610388	1.56664390	-1.08018415	-0.22201963	-0.443144578
sample8	1.08248149	0.71409337	-0.23929702	-1.27374339	-2.031347298
sample9	0.62214294	0.63795787	-0.36842320	-0.15035404	1.183472364
sample10	-2.46581394	0.16251259	-0.45722159	0.03614053	-0.709227630
sample11	-0.05473893	-0.91112765	-0.79030760	1.60522607	0.003912993
sample12	-0.26483767	-0.75149425	2.05662919	-0.42267057	0.629084883
sample13	0.20787970	0.13565926	-1.98594210	0.20316967	-1.242217646
sample14	-0.01808565	0.97796833	0.50376617	-0.15604184	-1.354532725
sample15	1.54267128	0.59218640	1.01732497	-0.46904671	-0.575065823

Gambar 5.9. Output data hasil bangkitan dari **Distribusi Normal Standar**

Pembangkitan data dari suatu distribusi statistika tertentu ini juga dapat dilakukan dengan command line di **R-Console**, yaitu dengan command **rnorm** (untuk Distribusi Normal) diikuti argumen optional yang diinginkan. Berikut adalah contoh pembangkitan data dengan command line untuk Distribusi Normal dengan rata-rata dan deviasi standar tertentu.

```
> rnorm(15, mean=0, sd=1)
[1] 0.66025751 -0.20716294 -1.03768624 -1.59951444 -0.09030604 -1.90549079
[7] -1.68778843 0.08368423 -0.96472623 -0.10300876 0.27261101 0.16491906
[13] 0.52697799 -0.57448961 -0.45865682

> latihan5 <- as.data.frame(matrix(rnorm(15*10, mean=0, sd=1), ncol=10))
> rownames(latihan5) <- paste("sample", 1:15, sep="")
> colnames(latihan5) <- paste("obs", 1:10, sep="")
> latihan5$mean <- rowMeans(latihan5[,1:10])
> showData(latihan5, placement='-20+200', font=getRcmdr('logFont'),
           maxwidth=80, maxheight=30)

> # Bangkitkan data dan simpan hasilnya dalam bentuk seperti matriks
> as.data.frame(matrix(rnorm(15*5, mean=100, sd=10), ncol=5))
```

	V1	V2	V3	V4	V5
1	84.46823	108.53078	104.05075	77.02379	91.55903
2	98.15929	93.74033	124.44052	80.38603	102.47690
3	95.00374	106.84794	104.09301	106.48609	97.34608
4	101.29297	118.54484	81.04212	98.63245	102.88233
5	98.92599	86.56266	86.52845	66.00474	90.27446
6	95.15418	102.50113	105.34845	79.55246	97.73824
7	106.38983	89.38471	85.31907	100.10805	91.51123
8	86.04483	104.22601	80.81650	101.08752	120.83886
9	84.41069	105.68604	91.14394	99.07307	99.37543
10	112.78286	104.58306	108.08592	109.01078	110.87053
11	109.17854	99.67204	97.54832	91.57182	104.02405
12	100.85442	98.14412	100.82436	97.54563	88.32492
13	111.41381	100.48431	103.03010	100.38959	101.00266
14	124.13427	101.54886	98.13771	102.57961	114.76246
15	93.99127	108.28097	107.97942	94.53939	86.20123
16	90.35201	123.02141	103.70384	95.25282	100.77538

Secara umum **R** menyediakan fasilitas untuk membangkitkan data dari berbagai distribusi statistika yang kontinu. Daftar lengkap berkaitan dengan command line di **R** untuk membangkitkan data dari distribusi kontinu beserta argumen dan library yang diperlukan dapat dilihat pada Tabel 5.1.

Tabel 5.1. Daftar **fungsi R (command line)** untuk membangkitkan data yang mengikuti suatu distribusi kontinu tertentu

Distribusi Kontinu	Fungsi R	Argumen yang diperlukan	library
Beta	rbeta	n, shape1, shape2	stats
Cauchy	rcauchy	n, location = 0, scale = 1	stats
Chi-squared	rchisq	n, df	stats
Ekspensial	rexp	n, rate	stats
F	rf	n, df₁, df₂	stats
Gamma	rgamma	n, shape, rate = 1	stats
Log-normal	rlnorm	n, mean, sd	stats
Logistic	rlogis	n, location = 0, scale = 1	stats
Normal	rnorm	n, mean, sd	stats
Student-t	rt	n, df	stats
Seragam (Uniform)	runif	n, min, max	stats
Weibull	rweibull	n, shape, scale = 1	stats
Multivariate Normal	mvrnorm	n = 1, mu, Sigma	MASS

5.2. Fungsi Distribusi Diskrit

Seperti pada bagian Distribusi Kontinu, pada bagian Fungsi Distribusi Diskrit ini akan dijelaskan cara perhitungan berkaitan dengan fungsi distribusi peluang, yaitu perhitungan nilai kuantil, pembuatan plot atau grafik densitas, pembuatan plot distribusi kumulatif, dan pembangkitan data dari suatu distribusi diskrit. Dalam hal ini, fokus pembahasan hanya diberikan pada Distribusi Binomial, sedangkan untuk distribusi diskrit yang lain dapat dilakukan dengan cara yang relatif sama.

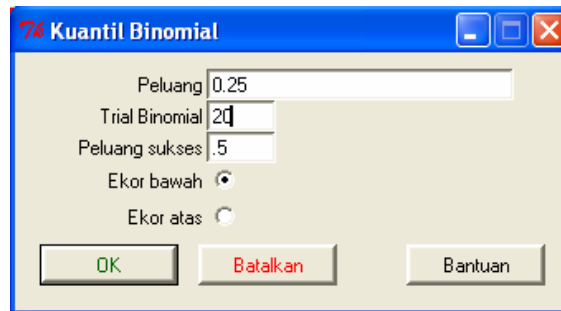
Secara umum, fungsi kepadatan probabilitas dari Distribusi Binomial adalah sebagai berikut

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x} \quad , \text{ untuk } x = \{0, 1, 2, \dots, n\}$$

dengan n adalah banyaknya pengamatan atau percobaan binomial, p adalah peluang sukses untuk suatu percobaan binomial, dan $(1-p)$ adalah peluang gagal atau tidak suksesnya. Notasi untuk peubah dan distribusinya adalah $X \sim B(n, p)$.

5.2.1. Menghitung Kuantil dari Distribusi Binomial

Perhitungan nilai kuantil tertentu dari Distribusi Binomial dapat dilakukan dengan **R-Commander**, yaitu gunakan menu **Distribusi**, pilih **Distribusi Diskrit**, pilih **Distribusi Binomial**, dan kemudian klik **Kuantil Binomial...**. Setelah itu akan terlihat jendela pilihan untuk mendapatkan kuantil yang akan dicari seperti pada Gambar 5.10.



Gambar 5.10. Jendela dialog untuk menghitung **Kuantil Binomial**

Misalkan akan dihitung nilai kuantil $\alpha=0,25$ (25%) dari **Distribusi Binomial** dengan $n=20$ dan $p=0.5$ atau $X \sim B(20,0.5)$, yaitu ingin dicari nilai X_α sedemikian hingga

$$P(X \leq X_\alpha) = 0,25 \quad (\text{luasan } \textit{lower tail} \text{ atau ekor bawah}).$$

Untuk mendapatkan kuantil di atas, maka pada jendela isian **Peluang** tulis nilai 0.25, **Trial Binomial** 20, dan **Peluang Sukses** 0.5. Kemudian klik **OK**, sehingga akan diperoleh nilai pada jendela keluaran **R-Commander** yaitu $X_{0,25} = 8$, yang berarti

$$P(X \leq 8) = 0,25.$$

Pilihan ekor atas (*upper tail*) digunakan jika akan dicari nilai $X_{1-\alpha}$ sedemikian hingga

$$P(X \leq X_{1-\alpha}) = 1 - \alpha \quad (\text{luasan } \textit{upper tail} \text{ atau ekor atas}).$$

Jika pilihan ekor atas yang digunakan, maka keluaran **R-Commander** memberikan nilai 12 pada jendela keluarannya, yang berarti

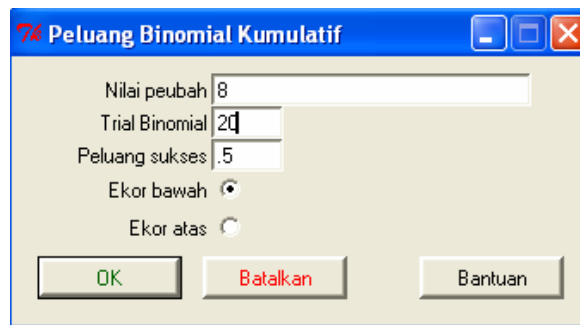
$$P(X \leq 12) = 0,75.$$

Perhitungan kuantil binomial dapat juga dilakukan dengan command line di **R-Console**, yaitu dengan command **qbinom** diikuti argumen optional yang diinginkan. Berikut adalah contoh perhitungan kuantil binomial dengan command line untuk $\alpha=0,25$ dan α yang lain.

```
> qbinom(c(0.25), size=20, prob=0.5, lower.tail=TRUE)
[1] 8
> qbinom(c(0.25), size=20, prob=0.5, lower.tail=FALSE)
[1] 12
> qbinom(c(0.75), size=20, prob=0.5, lower.tail=TRUE)
[1] 12
```

5.2.2. Menghitung Peluang dari Distribusi Binomial

Perhitungan peluang kumulatif untuk nilai tertentu dari Distribusi Binomial dapat dilakukan dengan **R-Commander**, yaitu gunakan menu **Distribusi**, pilih **Distribusi Diskrit**, pilih **Distribusi Binomial**, dan klik **Peluang ujung Binomial...**. Setelah itu akan terlihat jendela pilihan untuk memperoleh peluang yang dicari seperti Gambar 5.11.



Gambar 5.11. Jendela dialog untuk menghitung **Peluang Binomial Kumulatif**

Misalkan akan dihitung nilai peluang dari **Distribusi Binomial Kumulatif**, yaitu ingin dicari nilai $P(X \leq 8)$ (luasan *lower tail* atau ekor bawah) dari **Distribusi Binomial** dengan $n=20$ dan $p=0.5$, maka pada jendela isian **Nilai peubah** tulis nilai **8**. Dalam contoh ini isikan **Trial Binomial** 20, dan **Peluang Sukses** 0.5. Klik **OK**, sehingga akan diperoleh nilai 0.2517223 pada jendela keluaran **R-Commander**.

Selain itu, **R** juga memberikan fasilitas untuk menghitung nilai peluang untuk suatu nilai tertentu. Misalkan akan dicari $P(X = 8)$ dari **Distribusi Binomial** dengan $n=20$ dan $p=0.5$. Untuk itu, pilih menu **Distribusi**, pilih **Distribusi Diskrit**, pilih **Distribusi Binomial**, dan klik **Peluang Binomial...**. Isikan **Trial Binomial** 20, dan **Peluang Sukses** 0.5. Klik **OK**, maka akan ditampilkan nilai peluang untuk $X = 0, 1, 2, \dots, 20$.

Perhitungan peluang binomial dan peluang binomial kumulatif dapat juga dilakukan dengan command line di **R-Console**, yaitu dengan command **dbinom** (untuk peluang) dan **pbinom** (untuk peluang kumulatif) diikuti argumen optional yang diinginkan. Berikut adalah contoh perhitungan peluang binomial dan peluang binomial kumulatif dengan command line untuk nilai-nilai tertentu.

```
> dbinom(8, size=20, prob=0.5)
[1] 0.1201344

> pbinom(c(8), size=20, prob=0.5, lower.tail=TRUE)
[1] 0.2517223

> pbinom(c(8), size=20, prob=0.5, lower.tail=FALSE)
[1] 0.7482777

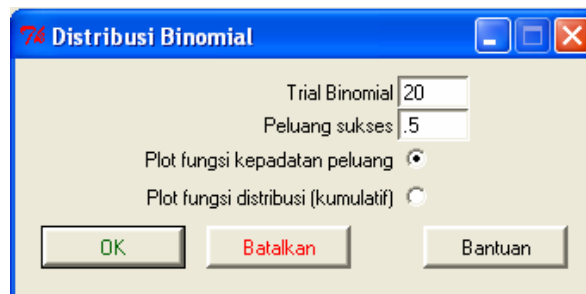
> pbinom(c(11), size=20, prob=0.5, lower.tail=FALSE)
[1] 0.2517223

> dbinom(0:20, size=20, prob=0.5)
[1] 9.536743e-07 1.907349e-05 1.811981e-04 1.087189e-03 4.620552e-03
[6] 1.478577e-02 3.696442e-02 7.392883e-02 1.201344e-01 1.601791e-01
[11] 1.761971e-01 1.601791e-01 1.201344e-01 7.392883e-02 3.696442e-02
[16] 1.478577e-02 4.620552e-03 1.087189e-03 1.811981e-04 1.907349e-05
[21] 9.536743e-07

> .Table <- data.frame(Pr=dbinom(0:20, size=20, prob=0.5))
> rownames(.Table) <- 0:20
> .Table
      Pr
0 9.536743e-07
1 1.907349e-05
2 1.811981e-04
3 1.087189e-03
4 4.620552e-03
5 1.478577e-02
6 3.696442e-02
... .....
16 4.620552e-03
17 1.087189e-03
18 1.811981e-04
19 1.907349e-05
20 9.536743e-07
```

5.2.3. Membuat Plot dari Distribusi Binomial

Plot dari Distribusi Binomial teoritis dengan **n** dan **p** tertentu dapat dilakukan dengan **R-Commander**, yaitu gunakan menu **Distribusi**, pilih **Distribusi Diskrit**, pilih **Distribusi Normal**, dan kemudian klik **Plot Distribusi Binomial...**. Setelah itu akan terlihat jendela pilihan untuk mendapatkan plot distribusi binomial teoritis yang ingin dicari seperti pada Gambar 5.12.



Gambar 5.12. Jendela dialog untuk membuat **Plot Distribusi Binomial**

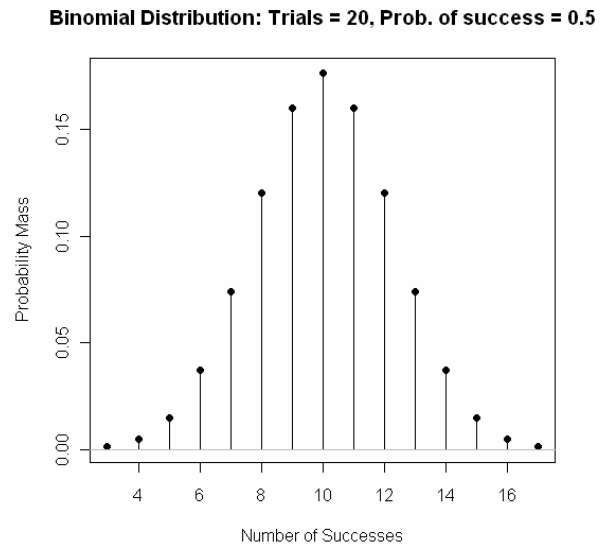
Misalkan akan dibuat plot fungsi kepadatan peluang dari **Distribusi Binomial** dengan **n=20** dan **p=0.5**, atau akan ditampilkan secara grafik nilai-nilai dari $f(x) = P(X = x)$ untuk $X \sim B(20, 0.5)$, atau

$$f(x) = \binom{20}{x} 0.5^x (1-0.5)^{20-x}, \text{ untuk } x = \{0, 1, 2, \dots, 20\}.$$

Untuk menampilkan itu, maka pada jendela tulis 20 pada isian **Trial Binomial**, dan tulis 0.5 pada isian **Peluang Sukses**.

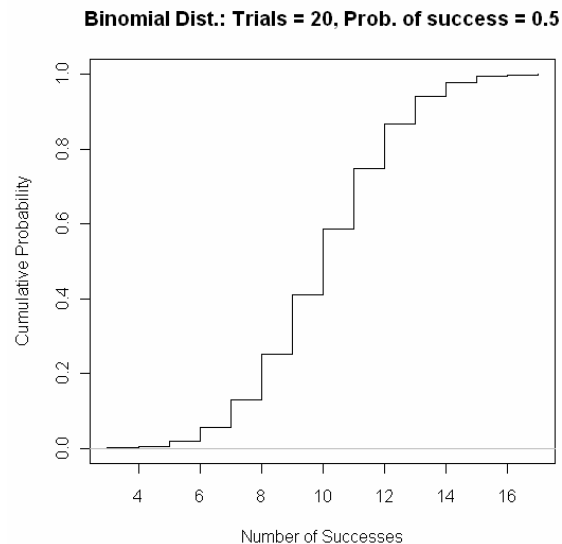
Setelah itu pilih plot yang akan dibuat, misalkan saja plot fungsi kepadatan peluang, maka klik pilihan **Plot fungsi kepadatan peluang**. Klik **OK**, sehingga akan diperoleh plot fungsi kepadatan dari Distribusi Binomial dengan **n=20** dan **p=0.5** seperti pada Gambar 5.13. Dari gambar ini dapat dilihat bahwa nilai $f(x)$ terbesar adalah pada $X = 10$, yang secara matematis dapat dihitung seperti berikut

$$\begin{aligned} f(10) &= \binom{20}{10} 0.5^{10} (1-0.5)^{20-10} \\ &= \binom{20}{10} 0.5^{10} (0.5)^{10} \\ &= 0.1601. \end{aligned}$$



Gambar 5.13. Output plot fungsi kepadatan **Distribusi Binomial** dengan **n=20** dan **p=0.5**

Jika pilihan **Plot fungsi distribusi (kumulatif)** atau $F(x) = P(X \leq x)$ yang dipilih, maka akan diperoleh output plot fungsi distribusi kumulatif dari Distribusi Binomial dengan **n=20** dan **p=0.5** seperti pada Gambar 5.14 berikut ini.



Gambar 5.14. Output plot fungsi **Distribusi Kumulatif Binomial** dengan **n=20** dan **p=0.5**

Pembuatan plot fungsi kepadatan dan fungsi distribusi kumulatif dapat juga dilakukan dengan command line di **R-Console**, yaitu dengan command **dnorm** (untuk plot fungsi kepadatan) dan command **pnorm** (untuk plot fungsi distribusi kumulatif) diikuti argumen optional yang diinginkan. Berikut adalah contoh pembuatan plot-plot tersebut dengan command line untuk suatu nilai peubah.

```
> # Perintah untuk pembuatan plot fungsi kepadatan binomial

> .x <- 3:17
> plot(.x, dbinom(.x, size=20, prob=0.5), xlab="Number of Successes",
      ylab="Probability Mass", main="Binomial Distribution: Trials = 20,
      Probability of success = 0.5", type="h")
> points(.x, dbinom(.x, size=20, prob=0.5), pch=16)
> abline(h=0, col="gray")

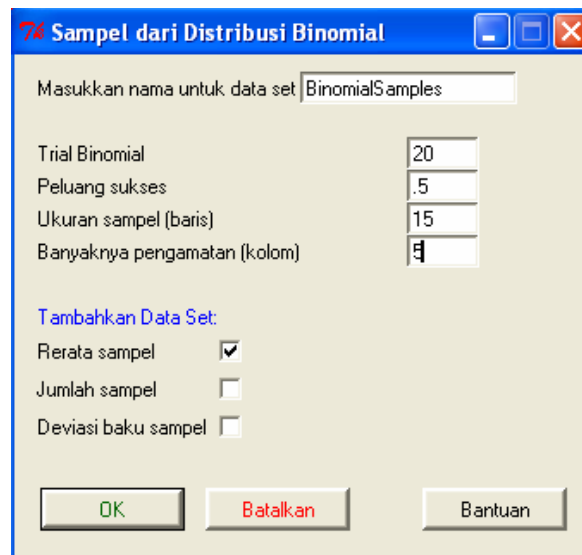
> # Perintah untuk pembuatan plot fungsi distribusi kumulatif binomial

> .x <- rep(.x, rep(2, length(.x)))
> plot(.x[-1], pbinom(.x, size=20, prob=0.5)[-length(.x)],
      xlab="Number of Successes", ylab="Cumulative Probability",
      main="Binomial Distribution: Trials = 20, Probability of success = 0.5",
      type="l")
> abline(h=0, col="gray")
```

5.2.4. Membangkitkan Data dari Distribusi Binomial

Seperti pada distribusi kontinu, **R** menyediakan fasilitas untuk membangkitkan data yang mengikuti distribusi diskrit tertentu. Misalkan akan dibangkitkan data yang mengikuti distribusi binomial, maka dapat digunakan menu **Distribusi**, pilih **Distribusi Diskrit**, pilih **Distribusi Binomial**, dan kemudian klik **Sampel dari Distribusi Binomial...**. Setelah itu akan terlihat jendela pilihan untuk pembangkitan data dari distribusi binomial seperti pada Gambar 5.15.

Sebagai contoh, akan dibangkitkan data sebanyak 15 baris dan 5 kolom yang mengikuti **Distribusi Binomial** dengan **n=20** dan **p=0.5**, maka tulis nama dataset hasil data bangkitan pada isian **Masukkan nama untuk data set** (misalkan **BinomialSamples**). Tulis angka 20 pada kolom isian **Trial Binomial**, dan angka 0.5 pada isian **Peluang Sukses**. Selanjutnya, pada pilihan **Ukuran sampel (baris)** ketik angka 15 dan 5 pada **Banyaknya pengamatan (kolom)**. Jika rata-rata sampel juga ingin ditambahkan, maka klik pada pilihan **Rerata sampel**, dan kemudian klik **OK**. Pilihan-pilihan yang lain, yaitu **Jumlah sampel** dan **Deviasi baku sampel** juga dapat ditampilkan jika diinginkan, yaitu dengan melakukan klik pada kedua pilihan tersebut.



Gambar 5.15. Jendela dialog untuk membangkitkan data dari **Distribusi Binomial**

Untuk mengetahui hasil data yang dibangkitkan, klik pilihan **Lihat data set** pada **R-Commander**, sehingga akan terlihat data-data hasil bangkitan seperti pada Gambar 5.16. Secara umum akan diperoleh 15 baris sampel dan 6 kolom data, yaitu 5 kolom data hasil bangkitan dan 1 kolom terakhir yang berisi rata-rata dari setiap sampel bangkitan.

	obs1	obs2	obs3	obs4	obs5	mean
sample1	12	9	9	6	10	9.2
sample2	10	9	11	13	11	10.8
sample3	14	13	13	14	9	12.6
sample4	12	9	10	10	6	9.4
sample5	10	6	7	10	9	8.4
sample6	8	10	12	15	9	10.8
sample7	9	12	14	12	10	11.4
sample8	7	10	9	10	9	9.0
sample9	8	10	14	10	7	9.8
sample10	8	12	11	9	6	9.2
sample11	11	10	7	6	10	8.8
sample12	12	6	12	7	13	10.0
sample13	7	11	10	9	8	9.0
sample14	11	10	13	7	9	10.0
sample15	9	10	8	6	11	8.8

Gambar 5.16. Output data hasil bangkitan dari **Distribusi Binomial** ($n=20$ dan $p=0.5$)

Proses pembangkitan data dari suatu distribusi statistika yang diskrit ini juga dapat dilakukan dengan command line di **R-Console**, yaitu dengan command **rbinom** (untuk Distribusi Binomial) diikuti argumen optional yang diinginkan. Berikut adalah contoh pembangkitan data dengan command line untuk Distribusi Binomial dengan **n** dan **p** tertentu.

```
> rbinom(100, size=20, prob=0.5)
[1] 12 12 9 12 13 6 8 8 7 11 11 7 10 8 9 12 9 9 10 10 8 12 8 9 9
[26] 11 8 12 12 11 13 15 6 11 11 12 8 10 11 9 8 11 12 8 13 10 14 12 12 11
[51] 12 11 11 12 11 11 7 17 6 12 9 6 11 10 7 8 8 11 9 10 8 7 10 11 6
[76] 14 9 12 9 9 7 10 12 11 14 12 12 13 13 3 12 12 14 10 10 8 6 9 15 15

> # Bangkitkan data binomial dan simpan hasilnya dalam matriks
> matrix(rbinom(15*5, size=20, prob=0.5), ncol=5)
      [,1] [,2] [,3] [,4] [,5]
[1,] 13 11 4 14 8
[2,] 11 10 12 8 10
[3,] 13 11 12 8 10
[4,] 12 8 11 9 9
[5,] 11 7 8 7 10
[6,] 11 11 10 11 14
[7,] 10 9 11 11 9
[8,] 11 12 12 7 12
[9,] 11 12 10 13 8
[10,] 12 8 12 11 6
[11,] 8 11 8 13 5
[12,] 13 11 9 12 8
[13,] 13 9 9 10 6
[14,] 10 9 11 12 10
[15,] 12 10 12 11 8

> BinomialSamples <- as.data.frame(matrix(rbinom(15*5, size=20, prob=0.5), ncol=5))
> rownames(BinomialSamples) <- paste("sample", 1:15, sep="")
> colnames(BinomialSamples) <- paste("obs", 1:5, sep="")
> BinomialSamples$mean <- rowMeans(BinomialSamples[,1:5])
> showData(BinomialSamples, placement='-20+200', font=getRcmdr('logFont'),
           maxwidth=80, maxheight=30)
```

Secara umum **R** menyediakan fasilitas untuk membangkitkan data dari berbagai distribusi statistika yang diskrit. Daftar lengkap berkaitan dengan command line di **R** untuk membangkitkan data dari distribusi diskrit beserta argumen dan **library** yang diperlukan dapat dilihat pada Tabel 5.2.

Tabel 5.2. Daftar **fungsi R (command line)** untuk membangkitkan data yang mengikuti suatu distribusi diskrit tertentu

Distribusi Diskrit	Fungsi R	Argumen yang diperlukan	library
Binomial	rbinom	n, size, prob	stats
Binomial Negatif	rnbinom	n, size, prob, mu	stats
Geometrik	rgeom	n, prob	stats
Hipergeometrik	rhyper	nn, m, n, k	stats
Poisson	rpois	n, lambda	stats

Berikut ini adalah ringkasan fungsi kepadatan probabilitas dari distribusi diskrit yang disediakan **R** pada tabel diatas.

▪ Distribusi **Binomial Negatif**

$$f(x) = \frac{\Gamma(x+n)}{\Gamma(n) x!} p^n (1-p)^x, \text{ untuk } x = \{0, 1, 2, \dots, n > 0\} \text{ dan } 0 < p \leq 1.$$

Distribusi ini merepresentasikan banyaknya kegagalan yang terjadi dalam suatu barisan percobaan Bernoulli sebelum suatu target dari sejumlah sukses dicapai.

▪ Distribusi **Geometrik**

$$f(x) = p(1-p)^{x-1}, \text{ untuk } x = \{1, 2, \dots\} \text{ dan } 0 < p \leq 1.$$

Distribusi ini merepresentasikan terjadinya sukses pertama kali pada percobaan ke x dalam suatu barisan percobaan Bernoulli.

▪ Distribusi **Hipergeometrik**

$$f(x) = \frac{\binom{m}{x} \binom{n}{k-x}}{\binom{m+n}{k}}, \text{ untuk } x = \{0, 1, 2, \dots, k\}.$$

Distribusi ini digunakan untuk sampling tanpa pengembalian. Fungsi kepadatan distribusi ini mempunyai parameter m (banyaknya objek group 1 yang berkaitan dengan banyaknya sukses), n (banyaknya objek group 2), dan k (banyaknya objek yang diambil tanpa pengembalian).

▪ Distribusi **Poisson**

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!}, \text{ untuk } x = \{1, 2, \dots\} \text{ dan } \lambda = \text{parameter rata-rata}.$$

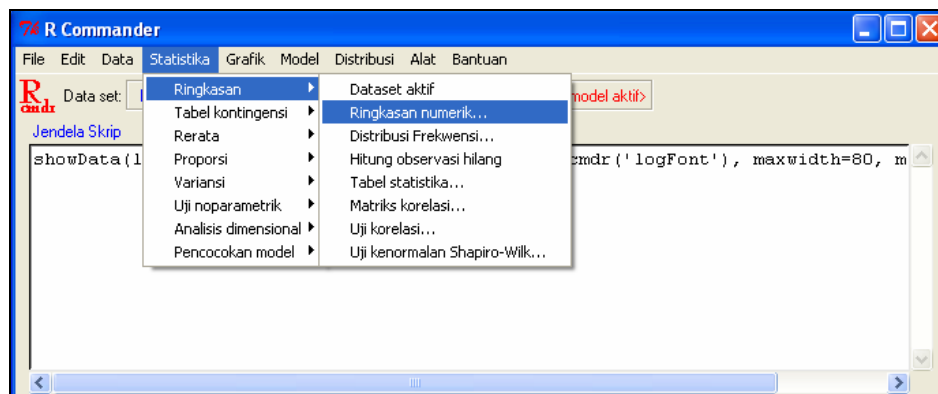
BAB 6

STATISTIK DESKRIPTIF MENGGUNAKAN R-Commander

Bab ini akan membahas penggunaan **R-Commander** untuk membuat statistik deskriptif dari suatu kumpulan data, khususnya pembuatan ringkasan (*summary*) data dan pembuatan tabel. Ringkasan data difokuskan pada pembuatan statistik deskriptif, yaitu ukuran-ukuran pemusatan, penyebaran, kemiringan, keruncingan, dan lokasi dari data-data numerik (metrik). Sedangkan pembuatan tabel difokuskan pada data-data nonnumerik (nonmetrik).

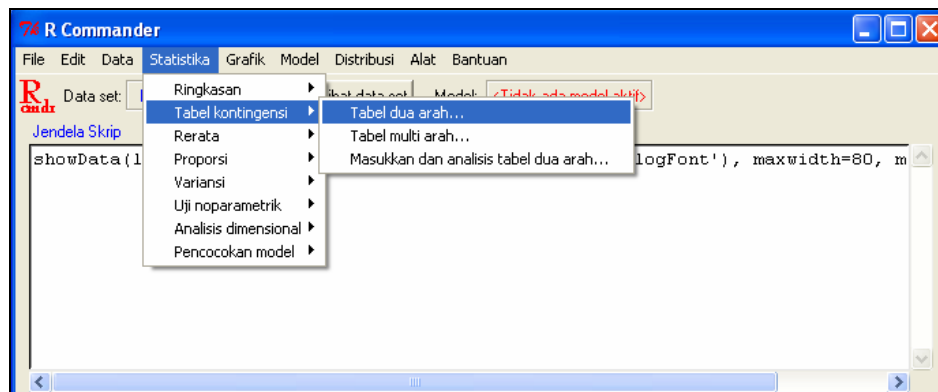
Paket **R** menyediakan beberapa fasilitas berkaitan dengan pembuatan ringkasan dari data numerik dan nonnumerik. Secara lengkap fasilitas yang berkaitan dengan ringkasan data dapat dilihat pada Gambar 6.1. Dari gambar ini dapat dilihat bahwa fasilitas Ringkasan data yang disediakan dalam **R-Commander** adalah

- Ringkasan numerik... ,
- Distribusi Frekuensi... ,
- Hitung observasi hilang
- Tabel statistika... ,
- Matriks korelasi... ,
- Uji korelasi... , dan
- Uji kenormalan Shapiro-Wilk... .



Gambar 6.1. Jendela dialog untuk pilihan **Distribusi Kontinu**

Selain tabel satu informasi, pada bab ini juga akan dijelaskan cara pembuatan tabel lebih dari satu informasi, khususnya tabel dua informasi secara bersama-sama yang dikenal dengan tabulasi silang atau tabel kontingensi. Fasilitas yang disediakan **R** untuk pembuatan tabel kontingensi ini dapat dilihat pada Gambar 6.2.



Gambar 6.2. Jendela dialog untuk pilihan **Distribusi Diskrit**

Sebagai langkah awal, buka kembali program **R** dengan mengklik icon **R 2.7.2.** dan panggil data tentang negara-negara di dunia pada tahun 1995 yang dikenal dengan data **WORLD95.SAV** di SPSS yang sudah disimpan dalam file **R** yaitu **latihan4.RData**, seperti yang digunakan pada Bab 4 sebelumnya. Load file workspace tersebut dengan menggunakan menu **File**, pilih **Load Workspace....**

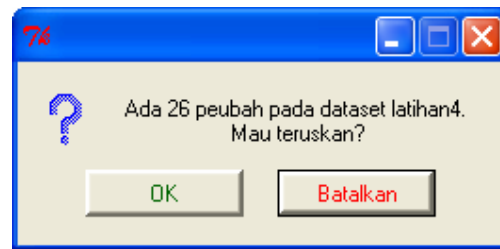
6.1. Ringkasan Numerik (Summary)

Pada bagian ini akan dijelaskan cara perhitungan ringkasan deskriptif dari data dengan menggunakan berbagai metode statistika deskriptif. Secara umum ada dua macam data yang akan dibuat ringkasan numeriknya, yaitu data metrik (skala interval atau rasio) dan data nonmetrik (skala nominal atau ordinal). **R** menyediakan dua macam cara untuk menampilkan ringkasan numerik dari variabel-variabel yang ada pada data, yaitu menampilkan ringkasan numerik dari semua variabel yang ada, dan menampilkan ringkasan numerik hanya dari variabel tertentu saja. Berikut ini adalah penjelasan lengkap untuk masing-masing cara pembuatan ringkasan numerik.

6.1.1. Ringkasan Numerik dari Semua Variabel

Perhitungan Ringkasan Numerik dari semua variabel dapat dilakukan dengan **R-Commander**, yaitu gunakan menu **Statistika**, pilih **Ringkasan**, pilih **Dataset aktif**. Setelah itu akan terlihat jendela informasi tentang jumlah variabel pada dataset yang akan ditampilkan ringkasan numeriknya seperti pada Gambar 6.3. Dalam hal ini, semua data baik yang metrik ataupun nonmetrik akan ditampilkan ringkasan numeriknya.

Pada data metrik, ringkasan numerik akan menampilkan beberapa besaran statistik yaitu **Mean**, **Min**, **Max**, **Kuartil 1**, **Median**, dan **Kuartil 3**. Sedangkan pada data nonmetrik, ringkasan numerik hanya menampilkan jumlah atau frekuensi pada masing-masing kategori yang ada.



Gambar 6.3. Jendela informasi jumlah variabel yang dibuat ringkasan numeriknya

Output lengkap yang diperoleh dari pilihan **Ringkasan** dan **Dataset aktif** pada data **latihan4.RData** beserta command line di **R-Console** adalah sebagai berikut.

```
> summary(latihan4)
```

COUNTRY		POPULATN		DENSITY		URBAN	
Afghanistan	: 1	Min.	: 256	Min.	: 2.3	Min.	: 5.00
Argentina	: 1	1st Qu.:	: 5100	1st Qu.:	: 29.0	1st Qu.:	: 40.75
Armenia	: 1	Median	: 10400	Median	: 64.0	Median	: 60.00
Australia	: 1	Mean	: 47724	Mean	: 203.4	Mean	: 56.53
Austria	: 1	3rd Qu.:	: 35600	3rd Qu.:	: 126.0	3rd Qu.:	: 75.00
Azerbaijan	: 1	Max.	: 1205200	Max.	: 5494.0	Max.	: 100.00
(Other)	: 103					NA's	: 1.00

RELIGION		LIFEEXPF		LIFEEXPM		LITERACY	
Catholic	: 41	Min.	: 43.00	Min.	: 41.00	Min.	: 18.00
Muslim	: 27	1st Qu.:	: 67.00	1st Qu.:	: 61.00	1st Qu.:	: 63.00
Protstnt	: 16	Median	: 74.00	Median	: 67.00	Median	: 88.00
Orthodox	: 8	Mean	: 70.16	Mean	: 64.92	Mean	: 78.34
Buddhist	: 7	3rd Qu.:	: 78.00	3rd Qu.:	: 72.00	3rd Qu.:	: 98.00
Animist	: 4	Max.	: 82.00	Max.	: 76.00	Max.	: 100.00
(Other)	: 6					NA's	: 2.00

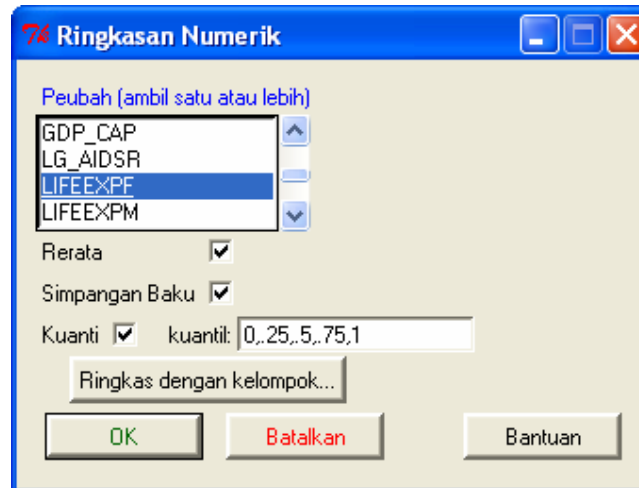
...

FERTILTY		LOG_POP		CROPGROW		LIT_MALE	
Min.	: 1.300	Min.	: 2.408	Min.	: 0.00	Min.	: 28.00
1st Qu.:	: 1.880	1st Qu.:	: 3.708	1st Qu.:	: 6.00	1st Qu.:	: 63.00
Median	: 3.050	Median	: 4.017	Median	: 13.50	Median	: 87.00
Mean	: 3.563	Mean	: 4.114	Mean	: 17.98	Mean	: 78.73
3rd Qu.:	: 5.000	3rd Qu.:	: 4.551	3rd Qu.:	: 26.75	3rd Qu.:	: 96.00
Max.	: 8.190	Max.	: 6.081	Max.	: 77.00	Max.	: 100.00
NA's	: 2.000			NA's	: 3.00	NA's	: 24.00

LIT_FEMA		CLIMATE	
Min.	: 9.00	temperate	: 34
1st Qu.:	: 45.00	tropical	: 32
Median	: 71.00	mediterranean	: 10
Mean	: 67.26	desert	: 7
3rd Qu.:	: 93.00	arid	: 6
Max.	: 100.00	(Other)	: 13
NA's	: 24.00	NA's	: 7

6.1.2. Ringkasan Numerik untuk Suatu Variabel

Perhitungan Ringkasan Numerik khusus untuk variabel metrik dapat dilakukan dengan **R-Commander**, yaitu gunakan menu **Statistika**, pilih **Ringkasan**, pilih **Ringkasan numerik...**. Setelah itu akan terlihat jendela informasi tentang variabel metrik dari dataset yang akan ditampilkan ringkasan numeriknya seperti pada Gambar 6.4 berikut.



Gambar 6.4. Jendela pilihan variabel metrik yang dibuat ringkasan numeriknya

Misalkan akan dibuat ringkasan numerik untuk variabel **LIFEEXPF** (usia harapan hidup wanita di suatu negara), maka pada jendela dialog yang muncul, klik **LIFEEXPF** pada pilihan **Peubah**. Kemudian klik besaran-besaran statistik yang akan ditampilkan ringkasannya. Setelah itu, klik **OK** untuk menampilkan output ringkasan numeriknya, sehingga diperoleh output pada jendela keluaran seperti berikut ini.

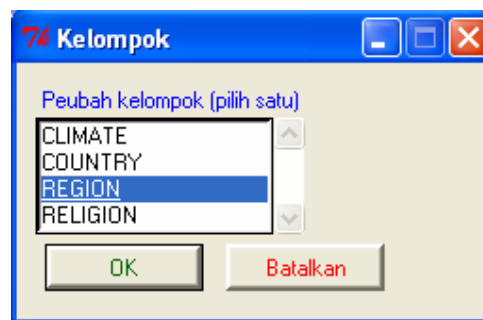
```
> numSummary(latihan4[, "LIFEEXPF"], statistics=c("mean", "sd", "quantiles"))
      mean      sd 0% 25% 50% 75% 100%   n
70.15596 10.57178 43  67  74  78  82 109
```

Perhitungan ringkasan numerik ini dapat juga dilakukan dengan command line di **R-Console**, yaitu dengan command **summary** diikuti argumen optional yang diinginkan. Berikut adalah contoh perhitungan ringkasan numerik dengan command line untuk suatu variabel metrik.

```
> summary(latihan4$LIFEEXPF)
```

```
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
43.00   67.00   74.00   70.16   78.00   82.00
```

R juga menyediakan fasilitas untuk menampilkan ringkasan numerik dari variabel numerik berdasarkan group atau faktor tertentu. Misalkan akan dibuat ringkasan untuk variabel **LIFEEXPF** berdasarkan **REGION**, maka gunakan menu **Statistika**, pilih **Ringkasan**, pilih **Ringkasan numerik...**, dan isikan pilihan seperti sebelumnya, yaitu variabel **LIFEEXPF** pada pilihan **Peubah** yang muncul. Kemudian klik **Ringkas dengan kelompok...**, dan pilih variabel **REGION** dari daftar **Peubah kelompok** yang ada seperti pada Gambar 6.5 berikut ini.



Gambar 6.5. Jendela pilihan peubah kelompok (group) dalam ringkasan numerik

Selanjutnya klik **OK**, maka akan diperoleh output ringkasan numerik pada jendela keluaran seperti pada output berikut ini.

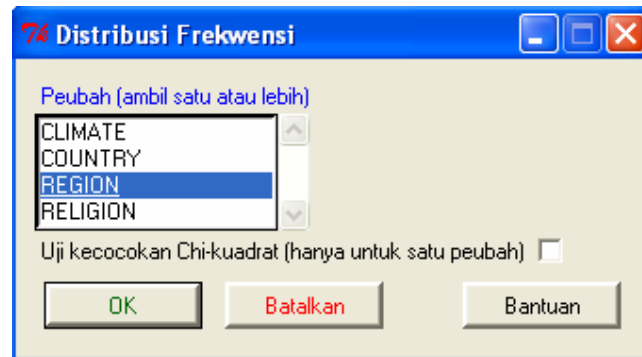
```
> numSummary(latihan4[,"LIFEEXPF"], groups=latihan4$REGION,
             statistics=c("mean", "sd", "quantiles"))
```

	mean	sd	0%	25%	50%	75%	100%	n
OECD	80.09524	1.179185	78	79	80	81	82	21
East Europe	76.00000	1.109400	74	75	76	77	78	14
Pacific/Asia	67.41176	10.886108	44	59	69	74	82	17
Africa	54.26316	7.978040	43	48	55	58	70	19
Middle East	71.58824	4.500817	63	68	72	74	80	17
Latn America	71.76190	7.388537	47	67	75	77	79	21

Dari output tersebut dapat dijelaskan bahwa rata-rata usia harapan hidup perempuan tertinggi pada negara-negara yang termasuk kawasan **OECD**, dan yang terendah adalah pada kawasan **AFRICA**. Secara visual hal ini seperti yang telah diperoleh pada **Plot Rata-rata** di Bab 4 sebelumnya.

6.2. Distribusi Frekuensi

Pembuatan Distribusi Frekuensi untuk variabel nonmetrik dapat dilakukan dengan **R-Commander**, yaitu gunakan menu **Statistika**, pilih **Ringkasan**, pilih **Distribusi Frekuensi...**. Setelah itu akan terlihat jendela pilihan tentang variabel nonmetrik dari dataset yang akan ditampilkan distribusi frekuensinya seperti pada Gambar 6.6.



Gambar 6.6. Jendela pilihan variabel nonmetrik (satu atau lebih) yang akan ditampilkan distribusi frekuensinya.

Misalkan akan dibuat distribusi frekuensi untuk variabel **REGION**, maka pada jendela dialog pilihan variabel yang muncul, klik **REGION** pada pilihan **Peubah**. Kemudian klik **OK** untuk menampilkan output distribusi frekuensinya, sehingga diperoleh output pada jendela keluaran seperti berikut ini.

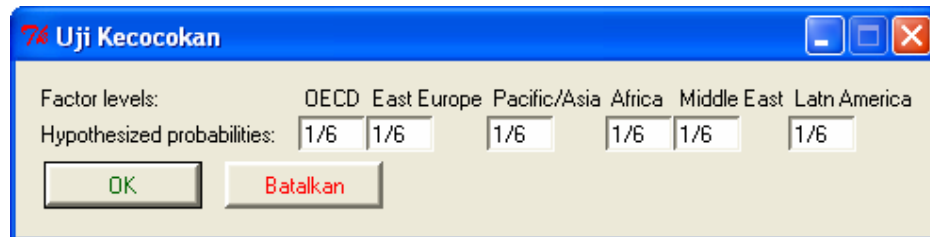
```
> .Table <- table(latihan4$REGION)
> .Table # counts for REGION

      OECD East Europe Pacific/Asia  Africa Middle East  Latn America
      21       14       17       19       17       21

> 100*.Table/sum(.Table) # percentages for REGION

      OECD East Europe Pacific/Asia  Africa Middle East  Latn America
19.26606 12.84404 15.59633 17.43119 15.59633 19.26606
```

Sebagai tambahan, pada menu pilihan **Distribusi Frekuensi...** ini dapat juga dilakukan Uji Kecocokan untuk mengevaluasi apakah probabilitas masing-masing kategori sesuai dengan yang dihipotesiskan. Misalkan akan diuji apakah persentase negara pada masing-masing **REGION** adalah sama, yaitu $1/6$, maka pilih **Uji kecocokan Chi-kuadrat (hanya untuk satu peubah)** sehingga diperoleh jendela pilihan probabilitas yang menjadi hipotesis seperti gambar berikut ini.



Gambar 6.7. Jendela pilihan isian probabilitas yang dihipotesiskan

Klik **OK** untuk menampilkan output hasil pengujian, sehingga diperoleh output pada jendela keluaran seperti berikut ini.

```
> .Probs <- c(0.166666666666667,0.166666666666667,0.166666666666667,
0.166666666666667,0.166666666666667,0.166666666666667)
```

```
> chisq.test(.Table, p=.Probs)
```

Chi-squared test for given probabilities

data: .Table

X-squared = 2.0275, df = 5, p-value = 0.8453

Hasil tersebut menunjukkan bahwa *p-value* pengujian adalah 0.8453. Sehingga jika digunakan $\alpha=0.05$ dapat disimpulkan bahwa pengujian menunjukkan gagal tolak H_0 . Hal ini dikarenakan *p-value* lebih besar dari α . Dengan demikian dapat disimpulkan bahwa proporsi negara di masing-masing **REGION** adalah **sama** yaitu $1/6$.

6.3. Tabel Statistika

R juga menyediakan fasilitas untuk membuat ringkasan statistik dalam tabel untuk suatu variabel numerik (metrik) berdasarkan variabel nonmetrik (kategorik) atau faktor tertentu. Pada **R-Commander**, gunakan menu **Statistika**, pilih **Ringkasan**, pilih **Tabel statistika...** sehingga diperoleh jendela pilihan seperti pada Gambar 6.8.



Gambar 6.8. Jendela pilihan Faktor dan Peubah respon yang akan ditampilkan Tabel Statistiknya

Misalkan akan dibuat tabel statistika untuk variabel **LOG_GDP** berdasarkan variabel **REGION**, maka pada jendela dialog pilihan **Faktor** yang muncul, klik **LOG_GDP** dan klik **REGION** pada pilihan **Peubah respon**. Kemudian pilih besaran statistik yang akan ditampilkan pada tabel statistika yang akan dibuat. Setelah itu, klik **OK** untuk menampilkan output tabel statistiknya, sehingga diperoleh output pada jendela keluaran seperti berikut ini.

```
> tapply(latihan4$LOG_GDP, list(REGION=latihan4$REGION), mean, na.rm=TRUE)

REGION
OECD East Europe Pacific/Asia Africa Middle East Latn America
4.207814 3.686428 3.107648 2.771881 3.546412 3.200901

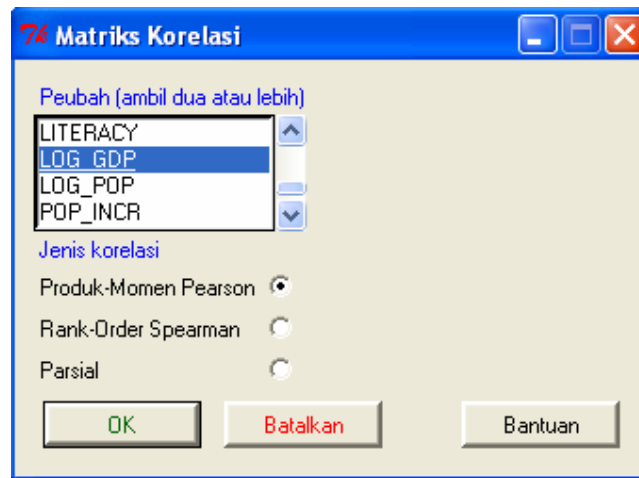
> tapply(latihan4$LIFEEXPF, list(REGION=latihan4$REGION), median, na.rm=TRUE)

REGION
OECD East Europe Pacific/Asia Africa Middle East Latn America
80 76 69 55 72 75
```

Hasil tersebut menunjukkan bahwa negara-negara di kawasan **OECD** mempunyai rata-rata **LOG-GDP** dan median **LIFEEXPF** paling tinggi dibanding dengan negara-negara di kawasan yang lain.

6.4. Matriks Korelasi

Perhitungan Matriks Korelasi untuk variabel-variabel metrik dan nonmetrik dapat dilakukan dengan **R-Commander**, yaitu gunakan menu **Statistika**, pilih **Ringkasan**, pilih **Matriks korelasi...**. Setelah itu akan terlihat jendela pilihan tentang variabel-variabel dari dataset yang akan ditampilkan matriks korelasinya seperti pada Gambar 6.9 berikut ini.



Gambar 6.9. Jendela pilihan Peubah yang akan ditampilkan matriks korelasinya

Secara umum, perhitungan nilai korelasi antara dua peubah metrik (skala *interval* atau *ratio*), misalkan X dan Y adalah (Johnson dan Bhattacharyya, 1996)

$$r_{xy} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

$$= \frac{\sum_{i=1}^n X_i Y_i - n\bar{X}\bar{Y}}{\sqrt{\sum_{i=1}^n X_i^2 - n\bar{X}^2} \sqrt{\sum_{i=1}^n Y_i^2 - n\bar{Y}^2}},$$

dengan X_i adalah nilai-nilai pada peubah pertama, Y_i adalah nilai-nilai pada peubah kedua, dan n adalah banyaknya pengamatan (data).

Misalkan akan dibuat matriks korelasi untuk variabel **CALORIES**, **FERTILITY**, **LIFEEXPF**, **LIFEEXPM**, dan **LOG_GDP**, maka pada jendela dialog pilihan **Peubah** yang muncul, klik kelima variabel tersebut. Kemudian pilih jenis korelasi (dalam kasus ini pilih **Produk-Momen Pearson**) yang akan ditampilkan pada matriks korelasi yang akan dibuat. Setelah itu, klik **OK** untuk menampilkan output matriks korelasi, sehingga diperoleh output pada jendela keluaran seperti berikut ini.

```
> cor(latihan4[,c("CALORIES","FERTILTY","LIFEEXPF","LIFEEXPM","LOG_GDP")],
      use="complete.obs")
```

	CALORIES	FERTILTY	LIFEEXPF	LIFEEXPM	LOG_GDP
CALORIES	1.0000000	-0.6958507	0.7753786	0.7650363	0.8474292
FERTILTY	-0.6958507	1.0000000	-0.8435988	-0.8089856	-0.7170879
LIFEEXPF	0.7753786	-0.8435988	1.0000000	0.9893717	0.8287739
LIFEEXPM	0.7650363	-0.8089856	0.9893717	1.0000000	0.8037349
LOG_GDP	0.8474292	-0.7170879	0.8287739	0.8037349	1.0000000

6.5. Uji Korelasi

Perhitungan Uji Korelasi, baik untuk korelasi **Produk-Momen Pearson** ataupun korelasi **Rank-Order Spearman** dapat dilakukan dengan **R-Commander**, yaitu gunakan menu **Statistika**, pilih **Ringkasan**, pilih **Uji korelasi...**. Setelah itu akan terlihat jendela pilihan tentang dua variabel yang akan diuji korelasinya seperti pada Gambar 6.10.



Gambar 6.10. Jendela pilihan dua Peubah yang akan diuji korelasinya

Misalkan akan dilakukan pengujian korelasi untuk variabel **LIFEEXPF** dan **LOG_GDP**, maka pada jendela dialog pilihan **Peubah** yang muncul, klik kedua variabel tersebut. Kemudian pilih jenis korelasi (dalam kasus ini pilih **Produk-Momen Pearson**) yang akan diuji. Setelah itu, klik **OK** untuk menampilkan output pengujian pada jendela keluaran seperti berikut ini.

```
> cor.test(latihan4$LIFEEXPF, latihan4$LOG_GDP, alternative="two.sided",
           method="pearson")
```

Pearson's product-moment correlation

data: latihan4\$LIFEEXPF and latihan4\$LOG_GDP

t = 15.4575, df = 107, p-value < 2.2e-16

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.7621177 0.8813950

sample estimates:

cor

0.8310795

Hipotesis yang digunakan dalam pengujian korelasi ini adalah sebagai berikut

$H_0 : \rho_{xy} = 0$ (kedua peubah tidak berkorelasi linear)

$H_1 : \rho_{xy} \neq 0$ (kedua peubah berkorelasi linear).

Statistik uji untuk pengujian korelasi ini adalah uji t , yaitu

$$t = \frac{r_{xy} \sqrt{n-2}}{\sqrt{1-r_{xy}^2}}.$$

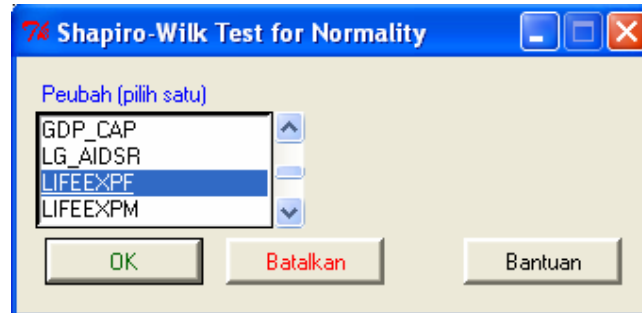
Dalam hal ini, H_0 ditolak yang berarti dua peubah secara statistik signifikan berkorelasi jika nilai uji t memenuhi daerah penolakan, yaitu

$$|t| > t_{\frac{\alpha}{2}; df=n-2} \quad \text{atau} \quad \text{nilai } p < \alpha.$$

Hasil output **R** diatas menunjukkan bahwa nilai **p** pengujian adalah 2.2e-16. Sehingga jika digunakan $\alpha=0.05$ dapat disimpulkan bahwa pengujian menunjukkan tolak H_0 . Hal ini dikarenakan nilai **p** lebih kecil dari α . Dengan demikian dapat disimpulkan bahwa ada korelasi positif antara **LIFEEXPF** dan **LOG-GDP**.

6.6. Uji Kenormalan Shapiro-Wilk

Perhitungan Uji Kenormalan Shapiro-Wilk pada **R-Commander** dapat dilakukan dengan menggunakan menu **Statistika**, pilih **Ringkasan**, kemudian pilih **Uji kenormalan Shapiro-Wilk...**. Setelah itu akan terlihat jendela pilihan tentang variabel yang akan diuji kenormalannya seperti pada Gambar 6.11 berikut ini.



Gambar 6.11. Jendela pilihan Peubah yang akan diuji kenormalannya

Hipotesis yang digunakan dalam pengujian kenormalan **Shapiro-Wilk** adalah sebagai berikut

- H_0 : data berdistribusi Normal, atau $X \sim N(\mu, \sigma^2)$
 H_1 : data tidak berdistribusi Normal.

Misalkan akan dilakukan pengujian kenormalan untuk variabel **LIFEEXPF**, maka pada jendela dialog pilihan **Peubah** yang muncul, klik **LIFEEXPF** tersebut. Kemudian klik **OK** untuk menampilkan output pengujian pada jendela keluaran seperti berikut ini.

```
> shapiro.test(latihan4$LIFEEXPF)

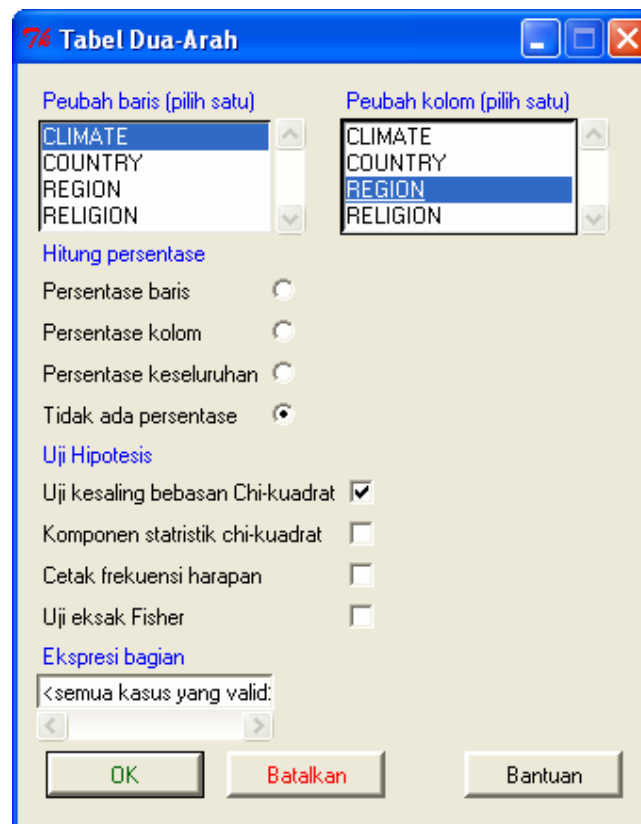
Shapiro-Wilk normality test

data: latihan4$LIFEEXPF
W = 0.8596, p-value = 9.435e-09
```

Output ini menunjukkan bahwa *p-value* pengujian adalah 9.435e-9. Sehingga jika digunakan $\alpha=0.05$ dapat disimpulkan bahwa pengujian menunjukkan tolak H_0 yang berarti data **LIFEEXPF** tidak berdistribusi normal. Hal ini didukung oleh QQ-Plot pada Bab 4 sebelumnya (lihat pada halaman 52).

6.7. Tabel Kontingensi Dua Arah

Pembuatan tabel kontingensi dua arah pada **R-Commander** dapat dilakukan dengan menggunakan menu **Statistika**, pilih **Tabel kontingensi**, dan kemudian pilih **Tabel dua arah...**. Setelah itu akan terlihat jendela pilihan tentang dua variabel nonmetrik (kategorik) yang akan dibuat tabel kontingensi dua arah seperti pada Gambar 6.12 berikut ini.



Gambar 6.12. Jendela pilihan Dua Peubah yang akan dibuat Tabel Kontingensinya

Misalkan akan dibuat tabel kontingensi dua arah untuk variabel **CLIMATE** dan **REGION**, maka pada jendela dialog pilihan **Peubah baris** yang muncul klik **CLIMATE** dan pilih **REGION** pada pilihan **Peubah kolom**. Kemudian pilih besaran (**persentase baris**, **persentase kolom**, atau **persentase keseluruhan**) dan uji hipotesis (sebagai pilihan default adalah **uji kesaling bebasan Chi-kuadrat**) yang akan dilakukan. Setelah itu, klik **OK** untuk menampilkan output tabel kontingensi dua arah pada jendela keluaran seperti berikut ini.

```

> .Table <- xtabs(~CLIMATE+REGION, data=latihan4)
> .Table

```

CLIMATE	REGION							
	OECD	East	Europe	Pacific/Asia	Africa	Middle	East	Latn America
desert	0		0		0	1	6	0
arid / desert	0		0		0	1	4	0
arid	1		0		1	0	2	2
tropical	0		0		9	10	0	13
mediterranean	1		2		2	2	2	1
maritime	1		3		0	0	0	0
temperate	15		8		4	2	2	3
arctic / temp	3		1		0	0	0	0
arctic	0		0		0	0	0	0

```

> rowPercents(.Table) # Row Percentages

```

CLIMATE	REGION							
	OECD	East	Europe	Pacific/Asia	Africa	Middle	East	Latn America
desert	0.0		0.0		0.0	14.3	85.7	0.0
arid / desert	0.0		0.0		0.0	20.0	80.0	0.0
arid	16.7		0.0		16.7	0.0	33.3	33.3
tropical	0.0		0.0		28.1	31.2	0.0	40.6
mediterranean	10.0		20.0		20.0	20.0	20.0	10.0
maritime	25.0		75.0		0.0	0.0	0.0	0.0
temperate	44.1		23.5		11.8	5.9	5.9	8.8
arctic / temp	75.0		25.0		0.0	0.0	0.0	0.0
arctic	NaN		NaN		NaN	NaN	NaN	NaN

```


```

CLIMATE	REGION	
	Total	Count
desert	100.0	7
arid / desert	100.0	5
arid	100.0	6
tropical	99.9	32
mediterranean	100.0	10
maritime	100.0	4
temperate	100.0	34
arctic / temp	100.0	4
arctic	NaN	0

```

> .Test <- chisq.test(.Table, correct=FALSE)

Warning in chisq.test(.Table, correct = FALSE) :
  Chi-squared approximation may be incorrect

> .Test

Pearson's Chi-squared test

data: .Table
X-squared = NaN, df = 40, p-value = NA

```

Output uji Chi-kuadrat atau *Chi-squared* tidak dapat diperoleh karena banyak nilai 0 pada beberapa sel kombinasi antara variabel **CLIMATE** dan **REGION**.

6.8. Entry Langsung Data Frekuensi untuk Tabel Kontingensi Dua Arah

R juga menyediakan fasilitas untuk membuat tabel kontingensi dua arah dengan cara memasukkan langsung frekuensi-frekuensi pada setiap kombinasi sel yang ada pada tabel kontingensi. Pembuatan masukkan tabel dua arah ini pada **R-Commander** dapat dilakukan dengan menggunakan menu **Statistika**, pilih **Tabel kontingensi**, dan kemudian pilih **Masukkan dan analisis tabel dua arah...**. Setelah itu akan terlihat jendela pilihan tentang **Banyaknya baris** dan **Banyaknya kolom**, serta **Masukkan frekuensi** yang akan dibuat tabel kontingensi dua arah seperti pada Gambar 6.13 berikut ini.

Masukkan Tabel Dua-Arah

Banyaknya baris: 2

Banyaknya kolom: 2

Masukkan frekuensi:

	h raga	netron
Pria	45	20
Wanita	25	40

Hitung persentase:

Persentase baris: ☐

Persentase kolom: ☐

Persentase keseluruhan: ☐

Tidak ada persentase: ☒

Uji Hipotesis:

Uji kesaling bebasan Chi-kuadrat: ☒

Komponen statistik chi-kuadrat: ☐

Cetak frekuensi harapan: ☐

Uji eksak Fisher: ☐

OK Batalkan Bantuan

Gambar 6.12. Jendela pilihan Dua Peubah yang akan dibuat Tabel Kontingensinya

Misalkan akan dibuat tabel kontingensi dua arah untuk variabel **PILIHAN ACARA TV** dan **GENDER RESPONDEN**, maka pada jendela kolom, tulis **Olah raga** dan **Sinetron**, dan tulis **Pria** dan **Wanita** pada jendela baris. Isikan angka 45, 20, 25, dan 40 pada empat sel isian yang ada (misal Pria cenderung menonton Olah raga, sedangkan Wanita cenderung menonton Sinetron). Kemudian klik **OK** untuk menampilkan output tabel kontingensi dua arah pada jendela keluaran seperti berikut ini.

```

> library(abind) # aktifkan terlebih dulu jika diperlukan

> .Table <- matrix(c(45,20,25,40), 2, 2, byrow=TRUE)
> rownames(.Table) <- c('Pria', 'Wanita')
> colnames(.Table) <- c('Olah raga', 'Sinetron')

> .Table # Counts
      Olah raga Sinetron
Pria      45      20
Wanita    25      40

> rowPercents(.Table)
      Olah raga Sinetron Total Count
Pria    69.2    30.8    100      65
Wanita   38.5    61.5    100      65

> .Test <- chisq.test(.Table, correct=FALSE)
> .Test

      Pearson's Chi-squared test

data: .Table
X-squared = 12.381, df = 1, p-value = 0.0004337

```

Prosedur uji *Chi-square Pearson* atau χ^2 pada output diatas (untuk evaluasi dependensi antara dua peubah non-metrik, skala *nominal* atau *ordinal*) adalah sebagai berikut (Johnson dan Bhattacharyya, 1996).

- (1). **Hipotesa :** H_0 : peubah pada baris dan kolom independen
 H_1 : peubah pada baris dan kolom dependen

- (2). **Statistik uji :**

$$\chi^2 = \sum_{i=1}^b \sum_{j=1}^k \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

dengan O_{ij} = jumlah pengamatan pada baris ke- i dan kolom ke- j ,

E_{ij} = nilai ekspektasi pengamatan pada baris ke- i dan kolom ke- j .

Perhitungan untuk nilai ekspektasi ini adalah sebagai berikut

$$E_{ij} = \frac{n_{i.} n_{.j}}{n},$$

dengan $n_{i.}$ = total pengamatan baris ke- i , $n_{.j}$ = total pengamatan kolom ke- j ,
 dan n = total pengamatan keseluruhan.

(3). **Daerah penolakan :**

Tolak H_0 yang berarti kedua peubah saling dependen (terkait) jika

$$\chi^2 > \chi^2_{\alpha, df=(b-1)(k-1)} \quad \text{atau} \quad \text{nilai } p < \alpha ,$$

dengan b = jumlah baris, dan k = jumlah kolom.

Hasil dari output di atas menunjukkan nilai uji *Chi-square Pearson* dan nilai **p** untuk pengambilan kesimpulan tentang ada tidaknya dependensi antara gender responden dan acara TV yang sering ditonton. Nilai uji *Chi-square Pearson* adalah **12.381**, dan nilai **p** sebesar **0.0004337**. Dengan demikian, jika digunakan $\alpha=0.05$ dapat disimpulkan bahwa pengujian menunjukkan tolak H_0 yang berarti bahwa gender responden dan acara TV yang ditonton tidak independen atau saling terkait. Keterkaitan dua variabel tersebut adalah Pria cenderung menonton Olah raga (**68,2% Pria**), sedangkan Wanita cenderung menonton Sinetron (**61,5% Wanita**).

BAB 7

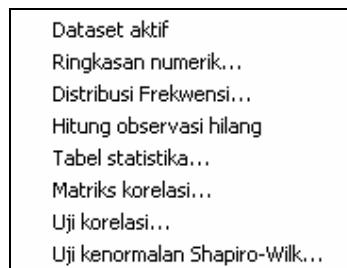
STATISTIK INFERENSI MENGGUNAKAN R-Commander

Pada bab ini akan dibahas penggunaan **R-Commander** untuk membuat analisis statistik inferensi dari suatu kumpulan data. Ada beberapa metode analisis statistik inferensi yang disediakan pada **R-Commander**, yaitu uji hipotesis untuk mean, proporsi, dan varians, uji Chi-kuadrat untuk evaluasi kebebasan antara dua variabel kategorik, uji ANOVA, uji-uji Nonparametrik, analisis komponen utama, analisis faktor, analisis klaster, analisis regresi linear, dan Generalized linear model.

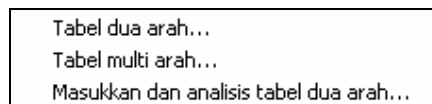
Paket **R-Commander** pada awalnya dibuat untuk keperluan analisis statistik yang sederhana, yaitu sebagai alat komputasi untuk perkuliahan statistika dasar, khususnya untuk pengguna yang cenderung lebih terbiasa menggunakan paket-paket statistika yang bersifat *point and click*. Oleh karena itu, menu dan pilihan kotak dialog yang ditampilkan masih bersifat sederhana dan tidak mencakup semua kapabilitas yang dimiliki **R**. Sebagai sebuah sistem komputasi statistika yang lengkap, kemampuan **R** sebagian besar diperoleh dari ribuan paket (*package* atau *library*) yang dikontribusikan oleh seluruh pengguna **R** di seluruh dunia. Dengan demikian, tidaklah mungkin membuat satu sistem **R-GUI** yang memiliki menu dari semua kemampuan yang dimiliki **R**. Hal ini karena terlalu banyaknya analisis statistika yang dapat dilakukan dengan menggunakan **R**. Untuk mengetahui daftar semua paket yang tersedia sampai saat ini dapat dilihat di <http://cran.r-project.org>.

Secara umum, metode statistika yang tersedia dalam **R-Commander** terbagi dalam 8 (delapan) dialog pilihan utama yang dapat dijalankan setelah memilih menu **Statistika**, yaitu :

1. Ringkasan (**Summaries**), yang terdiri dari dialog pilihan



2. Tabel kontingensi (**Contingency Tables**), yang terdiri dari dialog pilihan



3. Rerata (**Means**), yang terdiri dari dialog pilihan

Uji-t sampel tunggal...
Uji-t sampel saling bebas...
Uji-t berpasangan...
ANOVA Satu-arrah...
ANOVA Multi-arrah...

4. Proporsi (**Proportions**), yang terdiri dari dialog pilihan

Uji proporsi Sampel-tunggal...
Uji proporsi dua sampel...

5. Variansi (**Variances**), yang terdiri dari dialog pilihan

Uji-F Dua-variansi...
Uji Bartlett...
Uji Levene...

6. Uji nonparametrik (**Nonparametric tests**), yang terdiri dari dialog pilihan

Uji Wilcoxon Dua-sampel...
Uji Wilcoxon Sampel-berpasangan...
Uji Kruskal-Wallis...

7. Analisis dimensional (**Dimensional analysis**), yang terdiri dari dialog pilihan

Reliabilitas Skala...
Analisis Komponen Utama...
Analisis Faktor...
Analisis Kluster ▶

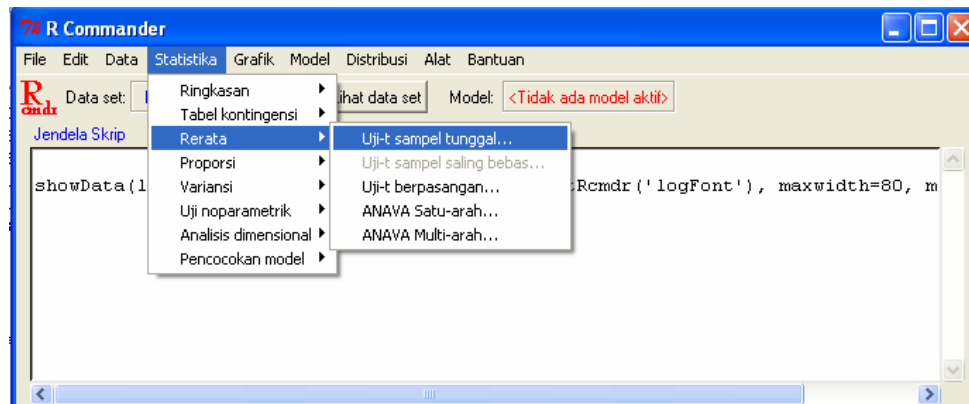
8. Pencocokan model (**Fit models**), yang terdiri dari dialog pilihan

Regresi Linier...
Model Linier...
Model Linier Tergeneralisir...
Model Multinomial logit...
model regresi ordinal...

Pembahasan tentang **Ringkasan** dan **Tabel kontingensi** sudah diberikan pada Bab 6 sebelumnya. Dengan demikian, bab ini akan membahas analisis statistika untuk pilihan **Rata-rata**, **Variansi**, dan seterusnya.

7.1. Pengujian Rata-rata (Mean)

Pada bagian ini akan dijelaskan cara perhitungan untuk pengujian rata-rata dari suatu data. R menyediakan lima macam pilihan pada pengujian rata-rata, yaitu **Uji-t sampel tunggal**, **Uji-t sampel saling bebas**, **Uji-t berpasangan**, **ANOVA Satu-arah**, dan **ANOVA Multi-arah**. Pilihan-pilihan analisis statistika tersebut dapat diperoleh dengan memilih menu **Statistika**, dan kemudian memilih **Rerata** seperti yang terlihat pada Gambar 7.1 berikut ini.



Gambar 7.1. Jendela dialog untuk pilihan pada **pengujian rata-rata**

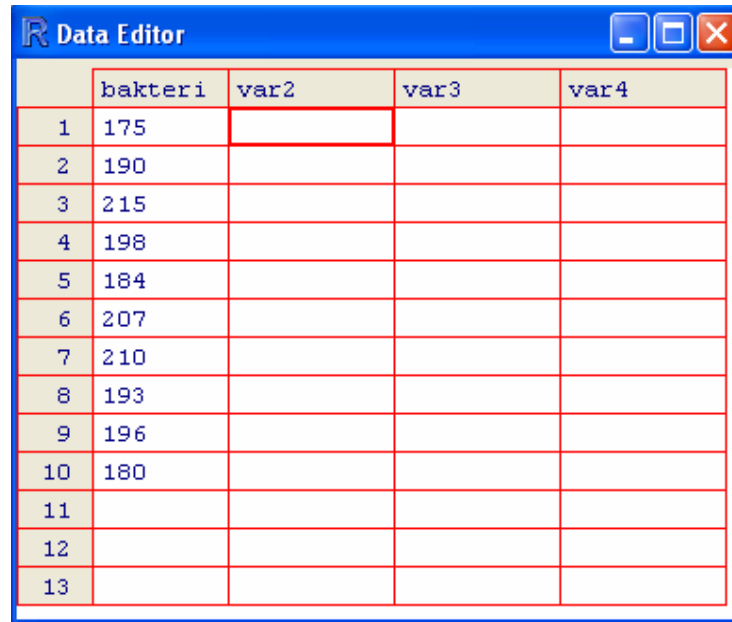
7.1.1. Pengujian Rata-rata sampel tunggal (Single sample t-test)

Misalkan suatu sampling terhadap air sungai KALIMAS Surabaya dilakukan oleh Departemen Kesehatan kota Surabaya untuk menentukan apakah rata-rata jumlah bakteri per unit volume air di Sungai tersebut masih di bawah ambang batas aman yaitu 200. Kemudian, peneliti di departemen tersebut mengumpulkan 10 sampel air per unit volume dan menemukan jumlah bakteri sebagai berikut.

Sampel ke	1	2	3	4	5	6	7	8	9	10
Jml. bakteri	175	190	215	198	184	207	210	193	196	180

Apakah data (informasi) ini memberikan bukti yang kuat bahwa rata-rata jumlah bakteri per unit volume air di sungai KALIMAS masih di bawah ambang batas aman?

Pengujian rata-rata sampel tunggal dapat dilakukan setelah data tersedia di **R**. Untuk itu, masukkan terlebih dahulu data-data tersebut dengan menggunakan menu **Data**, pilih **Dataset baru...**, dan beri nama dataset baru itu (misalkan **data7mu**). Setelah itu, isikan data-data itu seperti tampilan berikut ini.



	bakteri	var2	var3	var4
1	175			
2	190			
3	215			
4	198			
5	184			
6	207			
7	210			
8	193			
9	196			
10	180			
11				
12				
13				

Gambar 7.2. Jendela tampilan untuk entry data pada **R-Commander**

Untuk melakukan pengujian rata-rata sampel tunggal seperti contoh kasus di atas, **R-Commander** menyediakan pilihan yaitu melalui menu **Statistika**, pilih **Rerata**, dan kemudian pilih **Uji-t sampel tunggal...**, sehingga diperoleh tampilan dialog isian untuk pengujian rata-rata sampel tunggal seperti pada Gambar 7.3. Pada pilihan **Peubah** klik **bakteri**, dan kemudian isikan angka 200 pada kotak pilihan **Hipotesis nol: mu=**. Setelah itu, klik pilihan pada **Hipotesis Alternatif** sesuai dengan permasalahan diatas, yaitu **Rerata populasi < mu0**. Secara lengkap hipotesis statistik yang digunakan dalam pengujian rata-rata ini adalah sebagai berikut.

$$H_0 : \mu = 200 \text{ (atau } \mu \geq 200 \text{)}$$

$$H_1 : \mu < 200$$

Kemudian tentukan **Level Keyakinan** pengujian yang akan digunakan, misalkan saja 0.95. Hal ini berarti α yang digunakan adalah 5%. Setelah semua isian dialog sudah sesuai dengan pengujian yang akan dilakukan, klik **OK** untuk menampilkan output dari pengujian ini.



Gambar 7.3. Jendela dialog untuk pilihan pada **Uji-t sampel tunggal**

Statistik uji yang digunakan dalam uji hipotesis ini adalah uji t , yang rumus perhitungannya adalah (Johnson dan Bhattacharyya, 1996)

$$t = \frac{\bar{X} - \mu}{S / \sqrt{n}},$$

dengan \bar{X} adalah rata-rata dan S adalah deviasi standar yang dihitung dari data sampel, serta n adalah banyaknya data. Dalam hal ini, H_0 ditolak yang berarti bahwa $\mu < 200$, jika nilai uji t memenuhi daerah penolakan, yaitu

$$t < -t_{\alpha; df=n-1} \quad \text{atau} \quad p\text{-value} < \alpha.$$

Output hasil pengujian rata-rata sampel tunggal yang diperoleh dari contoh kasus di atas adalah sebagai berikut.

```
> t.test(data7mu1$bakteri, alternative='less', mu=200, conf.level=.95)
```

One Sample t-test

data: data7mu1\$bakteri

t = -1.2516, df = 9, p-value = 0.1211

alternative hypothesis: true mean is less than 200

95 percent confidence interval:

-Inf 202.4162

sample estimates:

mean of x

194.8

Hasil ini menunjukkan bahwa nilai statistik t yang diperoleh adalah -1.2516, dengan p -value pengujian adalah 0.1211. Dengan menggunakan kaidah pengambilan keputusan berdasarkan p -value, yaitu tolak H_0 jika p -value lebih kecil dari nilai α , maka pada $\alpha=0.05$ dapat disimpulkan bahwa pengujian menunjukkan gagal tolak H_0 . Dengan demikian dapat dijelaskan bahwa rata-rata jumlah bakteri per unit volume air yang ada di Sungai KALIMAS Surabaya tidak berada di bawah ambang batas aman atau pernyataan bahwa rata-rata jumlah bakteri per unit volume air di Sungai KALIMAS masih di bawah ambang batas aman adalah TIDAK BENAR.

7.1.2. Pengujian Perbedaan Rata-rata Dua sampel saling bebas atau *Independent sample t-test*

Misalkan suatu metode perakitan produk dalam pabrik tertentu memerlukan kira-kira satu bulan masa training untuk seorang pegawai baru untuk mencapai efisiensi maksimum. Suatu metode training yang baru telah diusulkan dan pengujian dilakukan untuk membandingkan metode baru tersebut dengan prosedur yang standar. Dua kelompok yang masing-masing terdiri dari sembilan pegawai baru dilatih selama periode waktu tiga minggu, satu kelompok menggunakan metode baru dan lainnya mengikuti prosedur latihan yang standar. Lama waktu (dalam menit) yang diperlukan oleh setiap pegawai untuk merakit produk dicatat pada akhir dari periode empat-minggu tersebut, dan hasilnya dapat dilihat pada tabel berikut.

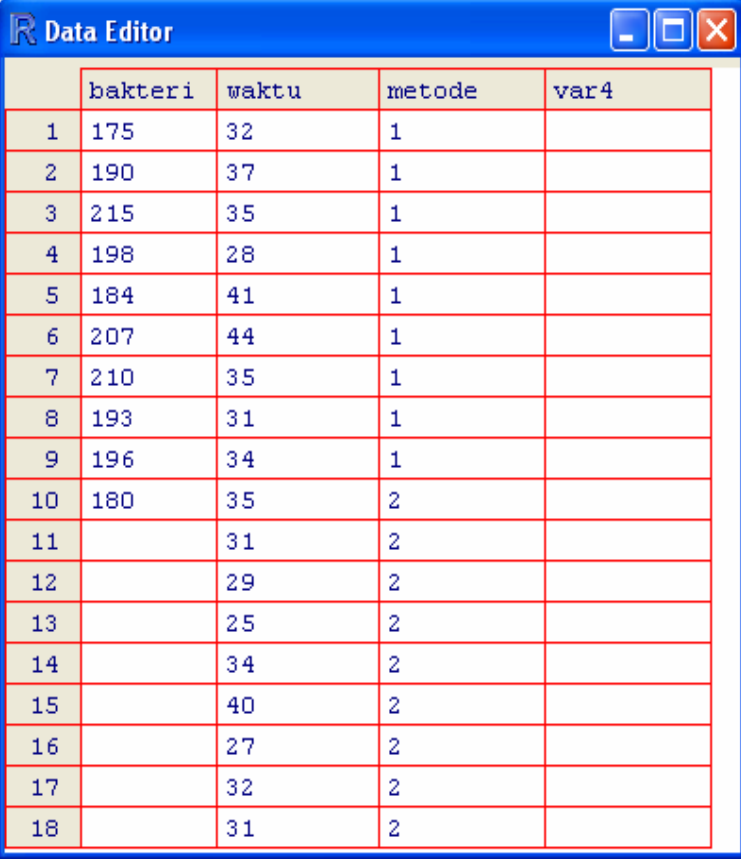
Tabel 7.1. Lama waktu (dalam menit) untuk merakit produk

Prosedur Standar	32	37	35	28	41	44	35	31	34
Prosedur Baru	35	31	29	25	34	40	27	32	31

Apakah data ini memberikan cukup bukti untuk menyatakan bahwa mean (rata-rata) waktu untuk merakit produk pada akhir periode empat minggu latihan adalah lebih kecil untuk prosedur (metode) latihan baru? Gunakan $\alpha=0.05$ untuk membuat kesimpulan dari pengujian hipotesis ini.

Seperti pada bagian sebelumnya, pengujian perbedaan rata-rata dua sampel independen ini dapat dilakukan setelah data tersedia di **R**. Dalam hal ini, ada dua cara yang dapat dilakukan yaitu memasukkan data pada dataset baru atau menambahkan data pada dataset yang sudah ada. Pada bagian ini akan digunakan cara kedua yaitu menambahkan data-data ini pada dataset yang sudah ada dari subbab sebelumnya, yaitu **data7mu**.

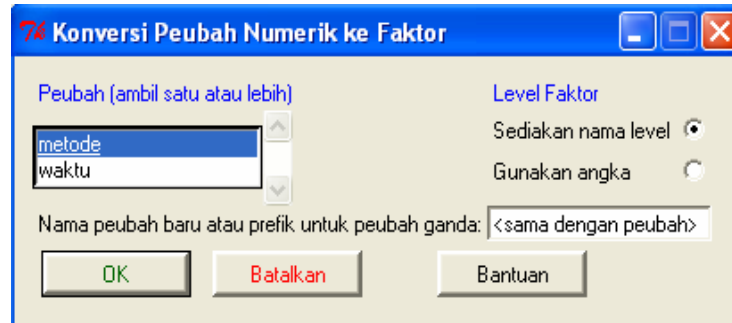
Untuk itu, aktifkan dulu **data7mu** dengan menggunakan menu **Data**, pilih **Dataset aktif**, dan kemudian klik **Pilih dataset aktif...**. Setelah itu pilih **data7mu** yang sudah tersimpan sebelumnya. Selanjutnya, editing data untuk menambah data baru dapat dilakukan dengan mengklik jendela dialog **Edit dataset**. Dengan demikian proses editing untuk menambahkan data baru dapat dilakukan. Isikan data-data pada Tabel 7.1 pada dua kolom baru yang tersedia, yaitu kolom pertama dengan nama **waktu** yang berisi data-data waktu perakitan (baik dengan metode baru ataupun metode standar). Sehingga pada kolom waktu ini ada 18 data. Pada kolom yang kedua beri nama **metode**, isikan angka-angka kode dari metode baru (misalkan dengan kode 1) dan metode standar (kode 2). Setelah proses input data baru telah lengkap, maka akan diperoleh tampilan dataset **data7mu** yang berisi 3 (tiga) kolom seperti yang terlihat pada Gambar 7.4. Kemudian tutup jendela pengisian data ini untuk melanjutkan ke komputasi pengujian perbedaan rata-rata dua sampel saling bebas.



	bakteri	waktu	metode	var4
1	175	32	1	
2	190	37	1	
3	215	35	1	
4	198	28	1	
5	184	41	1	
6	207	44	1	
7	210	35	1	
8	193	31	1	
9	196	34	1	
10	180	35	2	
11		31	2	
12		29	2	
13		25	2	
14		34	2	
15		40	2	
16		27	2	
17		32	2	
18		31	2	

Gambar 7.4. Jendela tampilan untuk hasil editing data baru pada **R-Commander**

Untuk dapat mengaktifkan menu **Uji-t sampel saling bebas** diperlukan langkah awal, yaitu mengkonversi variabel **metode** menjadi **faktor**. Hal ini dapat dilakukan dengan menggunakan menu **Data**, pilih **Atur peubah pada dataset aktif**, dan kemudian klik **Konversi peubah numerik ke faktor...**, sehingga diperoleh jendela dialog seperti gambar berikut ini.



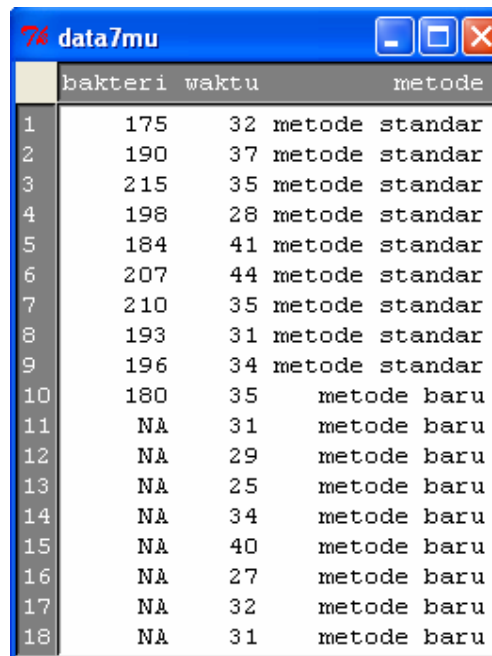
Gambar 7.5. Jendela dialog untuk **Konversi Peubah Numerik ke Faktor**

Selanjutnya pilih variabel **metode**, dan klik **Level Faktor** pada pilihan **Sediakan nama level** dan gunakan pilihan default **<sama dengan pubah>** pada **Nama peubah baru**. Klik **OK** sehingga diperoleh tampilan seperti berikut ini.



Gambar 7.6. Jendela dialog untuk **Nama level** pada peubah baru

Isikan nama level yang sesuai dengan nilai numerik yang akan diberi nama, yaitu **metode standar** untuk 1 dan **metode baru** untuk 2. Setelah itu klik **OK**, dan proses konversi variabel dari numerik ke faktor telah dilakukan. Untuk melihat perubahan data akibat proses konversi ini dapat dilakukan dengan mengklik pada jendela pilihan **Lihat data set**, sehingga diperoleh tampilan **data7mu** baru seperti pada Gambar 7.7 di bawah ini.



	bakteri	waktu	metode
1	175	32	metode standar
2	190	37	metode standar
3	215	35	metode standar
4	198	28	metode standar
5	184	41	metode standar
6	207	44	metode standar
7	210	35	metode standar
8	193	31	metode standar
9	196	34	metode standar
10	180	35	metode baru
11	NA	31	metode baru
12	NA	29	metode baru
13	NA	25	metode baru
14	NA	34	metode baru
15	NA	40	metode baru
16	NA	27	metode baru
17	NA	32	metode baru
18	NA	31	metode baru

Gambar 7.7. Jendela tampilan data baru setelah konversi **metode** ke **faktor**

Sebagai catatan, hasil editing dengan menambahkan variabel baru dengan jumlah data lebih banyak daripada variabel yang lama menyebabkan variabel yang lama mengandung data missing.

Tahap selanjutnya adalah proses pengujian perbedaan rata-rata untuk data di atas, yaitu dengan memilih menu **Statistika**, pilih **Rerata**, dan kemudian pilih **Uji-t sampel saling bebas...**, sehingga diperoleh jendela dialog seperti pada Gambar 7.8. Klik **metode** pada jendela **Kelompok**, dan **waktu** pada jendela **Peubah respon**, serta pilih **Hipotesis Alternatif** yang sesuai dengan permasalahan di atas, yaitu klik **Selisih > 0**. Dalam kasus ini, hipotesis statistika yang digunakan adalah

$$H_0 : \mu_1 - \mu_2 \leq 0 \text{ atau } \mu_1 \leq \mu_2$$

$$H_1 : \mu_1 - \mu_2 > 0 \text{ atau } \mu_1 > \mu_2$$

dengan μ_1 adalah rata-rata populasi untuk waktu merakit dengan prosedur standar, dan μ_2 menyatakan rata-rata populasi untuk waktu merakit dengan prosedur baru.

Setelah itu, pilih **Interval Keyakinan** yang digunakan (misalkan saja 0.95 yang berarti $\alpha=5\%$). Kemudian pilih **Asumsi variansi sama** dengan mengklik salah satu pilihan yang ada, misalkan saja **Ya** (pada bagian selanjutnya hal ini akan diuji dengan menggunakan fasilitas yang ada di **R-Commander**).



Gambar 7.8. Jendela dialog untuk Uji-t Sampel Saling Bebas

Penjelasan tentang statistik uji yang digunakan dalam uji hipotesis perbedaan dua rata-rata sampel saling bebas dengan asumsi variansi sama adalah uji t , yaitu (Johnson dan Bhattacharyya, 1996)

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{pooled} \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}},$$

dengan \bar{X}_1 dan \bar{X}_2 adalah rata-rata sampel pertama dan kedua, n_1 dan n_2 banyaknya sampel data pertama dan kedua, dan S_{pooled} adalah taksiran deviasi standar bersama yang didefinisikan dengan

$$S_{pooled}^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}.$$

Dalam hal ini S_1 dan S_2 adalah deviasi standar dari sampel data pertama dan kedua. Karena uji ini adalah uji satu arah dengan H_1 bertanda lebih besar, maka H_0 ditolak jika nilai uji t memenuhi daerah penolakan, yaitu

$$t > t_{\alpha; df=n_1+n_2-2} \quad \text{atau} \quad p\text{-value} < \alpha.$$

Selanjutnya, setelah semua isian dialog sudah sesuai dengan pengujian yang akan dilakukan, klik **OK** untuk menampilkan output dari pengujian ini. Hasil dari pengujian perbedaan rata-rata untuk kasus waktu merakit di atas secara lengkap dapat dilihat pada output berikut ini.

```
> fix(data7mu)
> data7mu$metode <- factor(data7mu$metode, labels=c('metode standar',
                                                    'metode baru'))

> t.test(waktu~metode, alternative='greater', conf.level=.95, var.equal=TRUE,
        data=data7mu)
```

Two Sample t-test

data: waktu by metode

t = 1.6495, df = 16, p-value = 0.05927

alternative hypothesis: true difference in means is greater than 0

95 percent confidence interval:

-0.2142871 Inf

sample estimates:

mean in group metode standar	mean in group metode baru
35.22222	31.55556

Hasil ini menunjukkan bahwa nilai statistik t yang diperoleh adalah 1.6495, dengan p -value sebesar 0.05927. Dengan menggunakan kaidah pengambilan keputusan berdasarkan p -value, maka pada $\alpha=0.05$ dapat disimpulkan bahwa pengujian hipotesis menunjukkan gagal tolak H_0 . Dengan demikian dapat dijelaskan bahwa rata-rata waktu perakitan dengan metode baru dan metode standar adalah tidak berbeda atau dugaan bahwa metode baru memberikan waktu perakitan lebih cepat adalah tidak didukung oleh data.

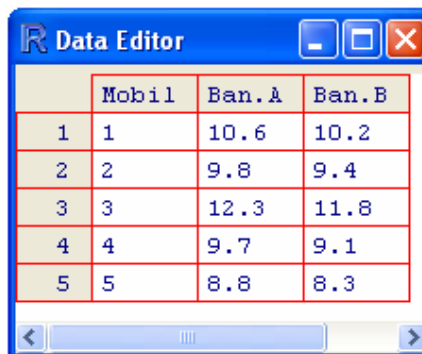
7.1.3. Pengujian Perbedaan Rata-rata sampel berpasangan (*Paired t-test*)

Misalkan sebuah pabrik ingin membandingkan kualitas keawetan dari dua jenis ban mobil yang berbeda, yaitu ban A dan B. Untuk perbandingan, dilakukan eksperimen dengan cara sebuah ban jenis A dan sebuah ban jenis B secara acak ditentukan dan dipasang pada roda belakang dari lima mobil. Mobil-mobil tersebut dijalankan untuk sejauh km tertentu dan jarak keawetan (jarak sampai diperoleh ban mengalami kerusakan tertentu) dicatat untuk setiap ban. Hasil pengukuran dari percobaan ini (dalam ribu km) dapat dilihat pada Tabel 7.2. Dalam percobaan ini, faktor pengemudi, kondisi mobil, kondisi jalan, dan faktor-faktor lain yang diduga berpengaruh terhadap tingkat keawetan pemakaian ban diharapkan dapat dikendalikan dengan cara melakukan pengacakan letak ban pada roda belakang setiap mobil yang digunakan. Berdasarkan data pada Tabel 7.2, tentukan apakah hasil ini memberikan cukup bukti untuk menyatakan bahwa ada perbedaan tingkat keawetan untuk kedua jenis ban mobil tersebut.

Tabel 7.2. Tingkat keawetan (dalam ribu km) untuk dua jenis ban

Mobil	Jenis Ban	
	Ban A	Ban B
1.	10,6	10,2
2.	9,8	9,4
3.	12,3	11,8
4.	9,7	9,1
5.	8,8	8,3

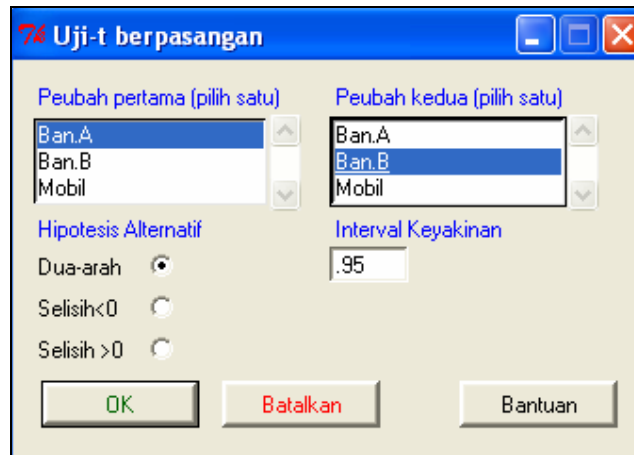
Pengujian perbedaan rata-rata sampel berpasangan dapat dilakukan dengan **R-Commander** setelah data tersedia di **R**. Untuk itu, aktifkan dulu **R-Commander** dan buat dataset baru, misalkan saja dengan nama **data7mu3** dengan menggunakan menu **Data**, pilih **Dataset baru...**. Setelah itu buat tiga kolom untuk variabel mobil, ban A, dan ban B. Isikan data pada Tabel 7.2 pada kolom-kolom baru yang tersedia, sehingga diperoleh tampilan data seperti pada Gambar 7.9 berikut ini.



	Mobil	Ban.A	Ban.B
1	1	10.6	10.2
2	2	9.8	9.4
3	3	12.3	11.8
4	4	9.7	9.1
5	5	8.8	8.3

Gambar 7.9. Jendela tampilan data untuk Uji-t berpasangan

Tahap selanjutnya adalah proses pengujian perbedaan rata-rata sampel berpasangan untuk data di atas, yaitu dengan memilih menu **Statistika**, pilih **Rerata**, dan kemudian pilih **Uji-t berpasangan...**, sehingga diperoleh jendela dialog seperti pada Gambar 7.10. Klik **ban.A** pada jendela **Peubah pertama**, dan **ban.B** pada jendela **Peubah kedua**. Kemudian pilih **Hipotesis Alternatif** yang sesuai dengan permasalahan di atas, yaitu klik **Dua-arah** yang menyatakan bahwa hipotesis penelitian adalah ada perbedaan tingkat keawetan antara ban A dan B.



Gambar 7.10. Jendela dialog untuk **Uji-t Berpasangan**

Dalam contoh kasus percobaan tingkat keawetan kedua ban ini, hipotesis statistika yang digunakan adalah

$$H_0 : \delta = 0$$

$$H_1 : \delta \neq 0 \text{ atau ada perbedaan tingkat keawetan}$$

dengan δ adalah rata-rata (populasi) selisih tingkat keawetan ban A dengan ban B. Statistik uji dalam pengujian ini adalah uji t , yaitu (Johnson dan Bhattacharyya, 1996)

$$t = \frac{\bar{D}}{S_D / \sqrt{n}},$$

dengan \bar{D} adalah rata-rata selisih sampel tingkat keawetan ban A dengan ban B, dan S_D adalah deviasi standar dari selisih sampel tingkat keawetan ban A dengan ban B, serta n adalah banyaknya sampel data. Karena uji ini termasuk dalam uji dua arah dengan H_1 bertanda tidak sama dengan atau \neq , maka H_0 ditolak jika nilai uji t memenuhi daerah penolakan, yaitu

$$|t| > t_{\frac{\alpha}{2}; df=n-1} \text{ atau } p\text{-value} < \alpha.$$

Kemudian, pilih **Interval Keyakinan** yang digunakan (misalkan saja 0.95 yang berarti $\alpha=5\%$). Setelah semua isian dialog sudah sesuai dengan pengujian yang akan dilakukan, klik **OK** untuk menampilkan hasil output dari pengujian sampel berpasangan seperti berikut ini.

```
> t.test(data7mu3$Ban.A, data7mu3$Ban.B, alternative='two.sided',
  conf.level=.95, paired=TRUE)
```

Paired t-test

data: data7mu3\$Ban.A and data7mu3\$Ban.B

t = 12.8285, df = 4, p-value = 0.0002128

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

0.3761149 0.5838851

sample estimates:

mean of the differences

0.48

Hasil ini menunjukkan bahwa nilai statistik t yang diperoleh adalah 12.8285, dan p -value pengujian adalah 0.0002128. Dengan menggunakan kaidah pengambilan keputusan berdasarkan p -value, maka pada $\alpha=0.05$ dapat disimpulkan bahwa pengujian menunjukkan tolak H_0 . Dengan demikian dapat dijelaskan bahwa rata-rata selisih tingkat keawetan antara ban A dan B adalah berbeda. Hasil ini menunjukkan bahwa ban A mempunyai tingkat keawetan lebih lama (jarak lebih jauh) dibanding ban B. Hal ini ditunjukkan oleh nilai positif pada rata-rata selisih jarak tempuh antara ban A dan B sampai ban-ban tersebut rusak.

7.1.4. Analisis Variansi (ANAVA) satu arah (*One-way ANOVA*)

Suatu eksperimen dilakukan untuk membandingkan harga sepotong roti (merek tertentu) pada empat lokasi di suatu kota. Empat toko pada lokasi 1, 2 dan 3 dipilih secara acak sebagai sampel, sedangkan di lokasi 4 hanya dua toko yang terpilih (hanya dua toko ini yang menjual merek tersebut). Diperoleh data sebagai berikut :

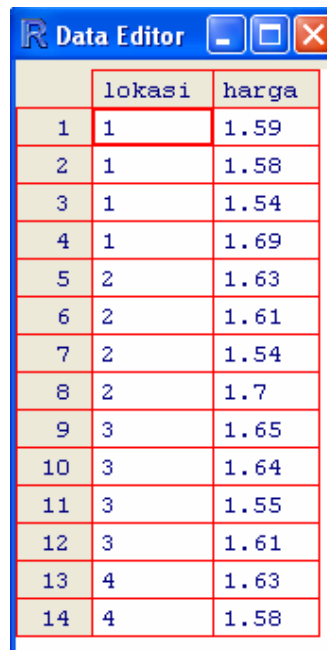
Tabel 7.3. Harga sepotong roti merek tertentu pada empat lokasi

Lokasi	Harga (ribu rupiah)			
1	1.59	1.63	1.65	1.61
2	1.58	1.61	1.64	1.63
3	1.54	1.54	1.55	1.58
4	1.69	1.70		

Apakah data ini memberikan bukti yang cukup untuk menyatakan bahwa ada perbedaan rata-rata harga roti di toko-toko pada 4 lokasi yang tersebar di kota tersebut?

Pengujian perbedaan rata-rata dari empat sampel ini dapat dilakukan dengan metode ANAVA satu arah yang tersedia di **R-Commander** setelah data tersedia di **R**. Untuk itu, buat dataset baru, misalkan saja dengan nama **data7mu4** dengan menggunakan menu **Data**, pilih **Dataset baru...**. Setelah itu buat dua kolom untuk variabel **lokasi**, dan **harga**. Isikan data pada Tabel 7.3 pada kolom-kolom baru yang tersedia, sehingga diperoleh data seperti pada Gambar 7.11.

Seperti pada **Uji-t sampel saling bebas**, diperlukan langkah awal untuk mengaktifkan **ANAVA satu arah** ini, yaitu mengkonversi variabel **lokasi** menjadi **faktor**. Hal ini dapat dilakukan dengan menggunakan menu **Data**, pilih **Atur peubah pada dataset aktif**, dan kemudian klik **Konversi peubah numerik ke faktor...**, seperti yang digunakan pada variabel **metode** pada **Uji-t sampel saling bebas** di bagian sebelumnya, yaitu sub-bab 7.1.2.



	lokasi	harga
1	1	1.59
2	1	1.58
3	1	1.54
4	1	1.69
5	2	1.63
6	2	1.61
7	2	1.54
8	2	1.7
9	3	1.65
10	3	1.64
11	3	1.55
12	3	1.61
13	4	1.63
14	4	1.58

Gambar 7.11. Jendela tampilan data untuk **ANAVA satu arah**

Dari gambar ini dapat dilihat bahwa struktur data yang digunakan adalah sama dengan pada pengujian rata-rata sampel saling bebas (independen).

Tahap selanjutnya adalah proses pengujian ANAVA satu arah, yaitu dengan memilih menu **Statistika**, pilih **Rerata**, dan kemudian pilih **ANAVA Satu-arah....** Selain itu, uji ANAVA satu arah ini dapat juga dilakukan dengan menggunakan menu **Statistika**, pilih **Rerata**, dan kemudian pilih **ANAVA Multi-arah...**, sehingga diperoleh jendela dialog seperti pada Gambar 7.12 berikut ini.



Gambar 7.12. Jendela dialog untuk ANAVA multi-arah

Pada jendela dialog pilihan **ANAVA multi-arah** terlihat bahwa fasilitas ini dapat digunakan untuk satu atau lebih **faktor**. Sehingga kalau hanya satu faktor yang diselidiki, maka fasilitas ini adalah sama saja dengan ANAVA satu arah.

Selanjutnya, klik **lokasi** pada jendela **Faktor**, dan **harga** pada jendela **Peubah respon**. Pada contoh kasus perbandingan rata-rata harga ini, hipotesis statistika yang digunakan adalah

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu$$

$$H_1 : \text{minimal ada satu mean populasi yang beda}$$

dengan μ_i adalah rata-rata (populasi) harga roti di lokasi i . Statistik uji yang digunakan adalah uji **F**, dan bentuk perhitungannya disajikan dalam suatu tabel yang dikenal dengan tabel ANAVA. Berikut ini adalah bentuk umum tabel ANAVA satu arah untuk perbandingan rata-rata k populasi (Johnson dan Bhattacharyya, 1996).

Sumber	d.f.	SS	MS	F
<i>Treatment</i>	$k - 1$	SST	$MST = SST / (k - 1)$	MST / MSE
<i>Error</i>	$n - k$	SSE	$MSE = SSE / (n - k)$	
<i>Total</i>	$n - 1$	<i>SS Total</i>		

Rumus untuk perhitungan nilai-nilai SST, SSE dan *SS Total* adalah sebagai berikut.

- Perhitungan SST atau *Sum Squares of Treatment*

$$SST = \sum_{i=1}^k n_i (\bar{x}_i - \bar{x})^2 = \sum_{i=1}^k \frac{T_i^2}{n_i} - \frac{T^2}{n}$$

- Perhitungan SSE atau *Sum Squares of Errors*

$$SSE = \sum_{i=1}^k \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2$$

- Perhitungan SS Total atau *Sum Squares of Total*

$$SS \text{ Total} = \sum_{i=1}^k \sum_{j=1}^{n_i} (x_{ij} - \bar{x})^2 = \sum_{ij} x_{ij}^2 - \frac{T^2}{n}$$

dengan

T_i = Total pengamatan populasi (*treatment*) ke- i

T = Total pengamatan seluruhnya

n_i = Banyaknya pengamatan *treatment* ke- i

n = Banyaknya pengamatan seluruhnya.

Kembali ke Gambar 7.12, setelah pengisian peubah pada jendela **Faktor**, dan jendela **Peubah respon**, selanjutnya klik **OK** untuk menampilkan output dari pengujian seperti berikut ini.

```
> Anova(lm(harga ~ lokasi, data=data7mu4))
```

Anova Table (Type II tests)

Response: harga

	Sum Sq	Df	F value	Pr(>F)
lokasi	0.000875	3	0.0897	0.964
Residuals	0.032525	10		

```
> tapply(data7mu4$harga, list(lokalasi=data7mu4$lokasi), mean, na.rm=TRUE) # means
```

lokasi

1	2	3	4
1.6000	1.6200	1.6125	1.6050

```
> tapply(data7mu4$harga, list(lokalasi=data7mu4$lokasi), sd, na.rm=TRUE) # std. deviations
```

lokasi

1	2	3	4
0.06377042	0.06582806	0.04500000	0.03535534

Hasil ini menunjukkan bahwa nilai statistik **F** yang diperoleh adalah 0.0897, dan *p-value* pengujian adalah 0.964. Dengan menggunakan kaidah pengambilan keputusan berdasarkan *p-value*, maka pada $\alpha=0.05$ dapat disimpulkan bahwa pengujian menunjukkan gagal tolak H_0 . Dengan demikian dapat dijelaskan bahwa rata-rata harga roti di empat lokasi itu adalah sama.

Selain dengan menggunakan perintah di **R-Commander**, dapat juga digunakan **command line** di **R-Console** yaitu menggunakan perintah **lm**. Perintah ini secara umum adalah untuk analisis model linear (**linear model**), termasuk juga dapat digunakan untuk analisis regresi linear. Berikut ini adalah contoh penggunaan perintah **lm** pada data di atas beserta outputnya.

```
> fit <- lm(harga ~ lokasi, data=data7mu4)
> summary(fit)
```

Call:
lm(formula = harga ~ lokasi, data = data7mu4)

Residuals:
 Min 1Q Median 3Q Max
 -0.08000 -0.02375 -0.00625 0.02687 0.09000

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
 (Intercept) 1.60000 0.02851 56.110 7.84e-14 ***
 lokasi[T.2] 0.02000 0.04033 0.496 0.631
 lokasi[T.3] 0.01250 0.04033 0.310 0.763
 lokasi[T.4] 0.00500 0.04939 0.101 0.921

 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05703 on 10 degrees of freedom
 Multiple R-Squared: 0.0262, Adjusted R-squared: -0.2659
 F-statistic: 0.08967 on 3 and 10 DF, p-value: 0.964

```
> anova(fit)
```

Analysis of Variance Table

Response: harga

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
lokasi	3	0.000875	0.000292	0.0897	0.964
Residuals	10	0.032525	0.003252		

Hasil dengan perintah ini memberikan output yang lebih banyak, termasuk koefisien regresi yang dapat digunakan untuk menghitung rata-rata harga roti pada setiap lokasi.

7.1.5. Analisis Variansi (ANOVA) dua arah (*Multi-way ANOVA*)

Suatu eksperimen dilakukan untuk menguji apakah terdapat efek (pengaruh) dari dua faktor, yaitu **jenis material** dan **temperatur** pemakaian, terhadap lama baterai tertentu dapat bertahan (usia pakai baterai). Tiga jenis material dan tiga macam temperatur berbeda dipilih secara acak dan digunakan dalam eksperimen ini. Dalam hal ini, temperatur yang dipilih adalah 15, 70, dan 125 (dalam $^{\circ}\text{F}$). Pada setiap sel kombinasi perlakuan dilakukan pengulangan (replikasi) sebanyak empat kali. Data hasil eksperimen ini secara lengkap dapat dilihat pada Tabel 7.4.

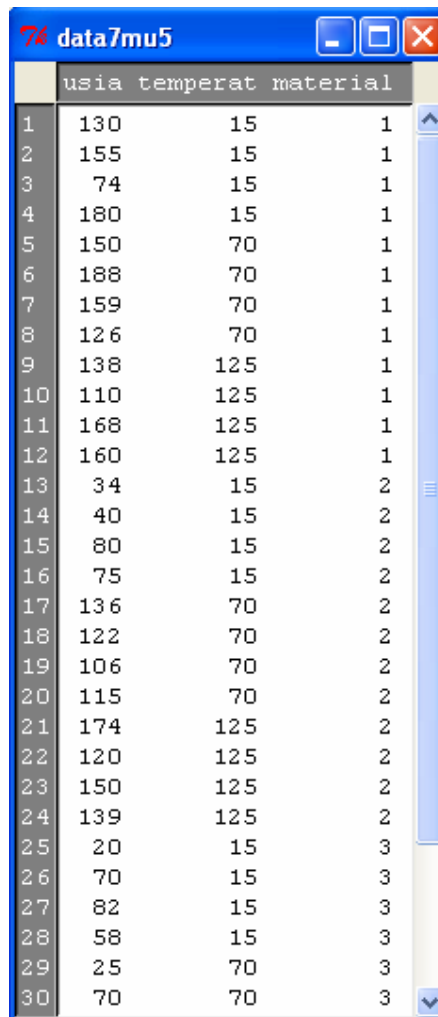
Tabel 7.4. Data eksperimen terhadap usia pakai baterai (dalam jam)

Tipe Material	Temperatur ($^{\circ}\text{F}$)					
	15		70		125	
1	130	155	34	40	20	70
	74	180	80	75	82	58
2	150	188	136	122	25	70
	159	126	106	115	58	45
3	138	110	174	120	96	104
	168	160	150	139	82	60

Apakah data ini memberikan bukti yang cukup untuk menyatakan bahwa ada efek **jenis material** dan **temperatur** terhadap **usia pakai baterai**?

Untuk menguji efek kedua faktor tersebut dapat dilakukan dengan metode ANOVA multi arah yang tersedia di **R-Commander**. Untuk itu, buat dataset baru, misalkan saja dengan nama **data7mu5** dengan menggunakan menu **Data**, pilih **Dataset baru...**. Setelah itu buat dua kolom untuk variabel **lokasi**, dan **harga**. Isikan data pada Tabel 7.4 pada kolom-kolom baru yang tersedia, sehingga diperoleh data seperti pada Gambar 7.13. Dari gambar ini dapat dilihat bahwa struktur data yang digunakan adalah sama dengan pada ANOVA satu arah.

Seperti pada **ANOVA satu arah**, diperlukan langkah awal untuk mengaktifkan **ANOVA satu arah** ini, yaitu mengkonversi variabel **jenis material** dan **temperatur** menjadi **faktor**. Hal ini dapat dilakukan dengan menggunakan menu **Data**, pilih **Atur peubah pada dataset aktif**, dan kemudian klik **Konversi peubah numerik ke faktor...**, seperti yang digunakan pada variabel **lokasi** pada **ANOVA satu arah** sebelumnya. Dalam kasus ini, karena ada dua faktor yang akan diselidiki pengaruhnya terhadap respon, maka analisis yang digunakan disebut **ANOVA dua arah**.



	usia	temperat	material	
1	130	15	1	
2	155	15	1	
3	74	15	1	
4	180	15	1	
5	150	70	1	
6	188	70	1	
7	159	70	1	
8	126	70	1	
9	138	125	1	
10	110	125	1	
11	168	125	1	
12	160	125	1	
13	34	15	2	
14	40	15	2	
15	80	15	2	
16	75	15	2	
17	136	70	2	
18	122	70	2	
19	106	70	2	
20	115	70	2	
21	174	125	2	
22	120	125	2	
23	150	125	2	
24	139	125	2	
25	20	15	3	
26	70	15	3	
27	82	15	3	
28	58	15	3	
29	25	70	3	
30	70	70	3	

Gambar 7.13. Jendela tampilan data untuk **ANAVA dua arah**

Tahap selanjutnya adalah proses pengujian ANAVA dua arah, yaitu dengan memilih menu **Statistika**, pilih **Rerata**, dan kemudian pilih **ANAVA Multi-arrah...**, sehingga diperoleh jendela dialog seperti pada Gambar 7.14. Pada jendela dialog pilihan **ANAVA multi-arrah** klik **material** dan **temperat** pada jendela **Faktor**, dan **usia** pada jendela **Peubah respon**. Pada contoh kasus ini, ada tiga hipotesis statistika yang digunakan, yaitu :

1. Efek faktor **Jenis Material**

$H_0 : \alpha_i = 0$ ($i=1,2,3$) atau tidak ada efek jenis material terhadap usia pakai

$H_1 : \text{minimal ada satu } \alpha_i \neq 0$ atau ada efek jenis material terhadap usia pakai

2. Efek faktor **Temperatur**

$H_0 : \beta_j = 0$ ($j=1,2,3$) atau tidak ada efek temperatur terhadap usia pakai

$H_1 : \text{minimal ada satu } \beta_j \neq 0$ atau ada efek temperatur terhadap usia pakai

3. Efek interaksi faktor **Jenis Material** dan **Temperatur**

$H_0 : (\alpha\beta)_{ij} = 0$ ($i,j=1,2,3$) atau tidak ada efek interaksi

$H_1 : \text{minimal ada satu } \beta_j \neq 0$ atau ada efek interaksi



Gambar 7.14. Jendela dialog untuk **ANAVA multi-arrah**

Kemudian, klik **OK** untuk menampilkan output dari pengujian **ANAVA dua-arrah** seperti dalam kotak di bawah paragraf ini. Hasil ini menunjukkan bahwa ada efek yang signifikan dari jenis material, temperatur, serta interaksi antara jenis material dan temperatur, terhadap usia pakai baterai. Hal ini ditunjukkan oleh *p-value* yang semuanya lebih kecil dari $\alpha=0.05$ pada ketiga efek yang dievaluasi.

```
> data7mu5$material <- as.factor(data7mu5$material)
> data7mu5$temperat <- as.factor(data7mu5$temperat)
> Anova(lm(usia ~ material*temperat, data=data7mu5))
```

Anova Table (Type II tests)

Response: usia

	Sum Sq	Df	F value	Pr(>F)
material	39119	2	28.9677	1.909e-07 ***
temperat	10684	2	7.9114	0.001976 **
material:temperat	9614	4	3.5595	0.018611 *
Residuals	18231	27		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

> tapply(data7mu5$usia, list(material=data7mu5$material,
  temperat=data7mu5$temperat), mean, na.rm=TRUE) # means

      temperat
material      15      70     125
1 134.75 155.75 144.00
2  57.25 119.75 145.75
3  57.50  49.50  85.50

> tapply(data7mu5$usia, list(material=data7mu5$material,
  temperat=data7mu5$temperat), sd, na.rm=TRUE) # std. deviations

      temperat
material      15      70     125
1 45.35324 25.61738 25.97435
2 23.59908 12.65899 22.54440
3 26.85144 19.26136 19.27866

> tapply(data7mu5$usia, list(material=data7mu5$material,
  temperat=data7mu5$temperat), function(x) sum(!is.na(x))) # counts

      temperat
material 15 70 125
1      4  4  4
2      4  4  4
3      4  4  4

```

Sebagai tambahan, hasil dari perintah **ANAVA multi arah** pada **R-Commander** juga menampilkan output yang berisi nilai-nilai mean, standar deviasi, dan jumlah pengulangan pada setiap sel kombinasi antar faktor.

7.2. Pengujian Kesamaan Variansi

R menyediakan tiga macam pilihan pada pengujian kesamaan variansi, yaitu **Uji-F dua variansi**, **Uji Bartlett**, dan **Uji Levene**. Pilihan-pilihan analisis statistika tersebut dapat diperoleh dengan memilih menu **Statistika**, dan kemudian memilih **Variansi**. Berikut ini adalah penjelasan untuk masing-masing uji tersebut.

7.2.1. Pengujian Kesamaan Dua Variansi

Pada bagian 7.1.2 sebelumnya telah dibahas pengujian rata-rata dua sampel independen dengan uji **t**. Dalam uji ini ada dua pilihan berkaitan dengan asumsi variansi dari dua sampel yang diamati, yaitu sama atau berbeda. Untuk menguji kesamaan dua variansi tersebut dapat dilakukan dengan uji **F**.

Perhatikan kembali contoh kasus perbandingan lama waktu merakit produk antara metode standar dan metode baru yang datanya dapat dilihat di Tabel 7.1. Untuk melakukan pengujian kesamaan variansi lama waktu merakit pada kedua metode tersebut, **R** menyediakan fasilitas dengan cara memilih menu **Statistika**, pilih **Variansi**, dan setelah itu pilih **Uji-F Dua-variansi...**, sehingga diperoleh jendela dialog seperti pada Gambar 7.15. (Aktifkan terlebih dahulu **dataset** yang sudah tersimpan sebelumnya, yaitu **dataset** pada bagian 7.1.2)



Gambar 7.15. Jendela dialog untuk **Uji-F Dua Variansi**

Selanjutnya, klik **metode** pada jendela **Kelompok**, dan **waktu** pada jendela **Peubah respon**. Pada contoh kasus perbandingan dua variansi dari lama waktu merakit produk dengan metode standar dan metode baru, hipotesis statistika yang digunakan adalah (Johnson dan Bhattacharyya, 1996)

$$H_0 : \sigma_1^2 = \sigma_2^2 \text{ atau kedua variansi adalah sama besar}$$

$$H_1 : \sigma_1^2 \neq \sigma_2^2 \text{ atau kedua variansi adalah berbeda.}$$

Statistik uji yang digunakan adalah uji **F**. Ada tiga pilihan **Hipotesis Alternatif** atau H_1 yang dapat dilakukan, yaitu **Dua-arah**, **Selisih<0**, dan **Selisih>0**. Pada contoh ini klik pilihan **Dua-arah** sesuai dengan yang dinyatakan pada hipotesis statistika di atas. Setelah itu tetapkan **Level Keyakinan** yang digunakan dalam pengujian (misalkan 0,95 yang berarti $\alpha=0.05$). Klik **OK** sehingga diperoleh tampilan output pada jendela keluaran seperti berikut ini.

```
> var.test(waktu ~ metode, alternative='two.sided', conf.level=.95, data=data7mu2)
```

F test to compare two variances

data: waktu by metode

F = 1.2205, num df = 8, denom df = 8, p-value = 0.7849

alternative hypothesis: true ratio of variances is not equal to 1

95 percent confidence interval:

0.2753114 5.4109136

sample estimates:

ratio of variances

1.220527

Hasil ini menunjukkan bahwa tidak ada perbedaan varians dari lama waktu merakit produk dengan metode standar dan metode baru. Hal ini ditunjukkan oleh **p-value** (yaitu 0.7849) yang lebih besar dari $\alpha=0.05$.

7.2.2. Uji Bartlett

R menyediakan fasilitas untuk pengujian kesamaan varians dari beberapa sampel (lebih dari dua sampel). Sebagai contoh kasus, lihat kembali bagian 7.1.5 tentang pengujian tentang efek jenis material dan temperatur terhadap usia pakai baterai. Misalkan ingin diketahui apakah ada perbedaan varians usia pakai baterai pada ketiga jenis material yang digunakan, maka dapat digunakan menu **Statistika**, pilih **Variansi**, dan setelah itu pilih **Uji Bartlett...**, sehingga diperoleh jendela dialog seperti pada Gambar 7.16. (Aktifkan terlebih dahulu **dataset** yang sudah tersimpan sebelumnya, yaitu **data7mu5** pada bagian 7.1.5) .



Gambar 7.16. Jendela dialog untuk **Uji Bartlett**

Selanjutnya, pilih **material** pada jendela **Kelompok**, dan **usia** pada jendela **Peubah respon**. Pada contoh kasus ini, hipotesis statistika yang digunakan adalah

$$H_0 : \sigma_1^2 = \sigma_2^2 = \sigma_3^2 \text{ atau ketiga varians adalah sama besar}$$

H_1 : minimal ada satu varians yang berbeda.

Klik **OK** sehingga diperoleh tampilan output pada jendela keluaran seperti berikut ini.

```
> bartlett.test(usia ~ material, data=data7mu5)
```

Bartlett test of homogeneity of variances

data: usia by material

Bartlett's K-squared = 2.8321, df = 2, p-value = 0.2427

Hasil ini menunjukkan bahwa tidak ada perbedaan varians dari usia pakai baterai pada ketiga jenis material yang digunakan dalam eksperimen. Hal ini ditunjukkan oleh *p-value* (yaitu 0.2427) yang lebih besar dari $\alpha=0.05$.

7.2.3. Uji Levene

Seperti pada bagian sebelumnya, **Uji Levene** adalah uji yang dapat digunakan untuk pengujian kesamaan varians dari beberapa sampel (lebih dari dua sampel). Perhatikan kembali contoh pada bagian sebelumnya, yaitu apakah ada perbedaan varians usia pakai baterai pada ketiga jenis material yang digunakan. **R** menyediakan fasilitas untuk **Uji Levene** yaitu melalui menu **Statistika**, pilih **Variansi**, dan setelah itu pilih **Uji Levene...**, sehingga diperoleh jendela dialog seperti pada Gambar 7.16.



Gambar 7.17. Jendela dialog untuk **Uji Levene**

Selanjutnya, pilih **material** pada jendela **Kelompok**, dan **usia** pada jendela **Peubah respon**. Seperti pada contoh sebelumnya, hipotesis statistika yang digunakan adalah

$$H_0 : \sigma_1^2 = \sigma_2^2 = \sigma_3^2 \text{ atau ketiga varians adalah sama besar}$$

$$H_1 : \text{minimal ada satu varians yang berbeda.}$$

Klik **OK** sehingga diperoleh tampilan output pada jendela keluaran seperti berikut ini.

```
> tapply(data7mu5$usia, data7mu5$material, var, na.rm=TRUE)

      1      2      3
1004.5152 1838.9924 659.0606

> levene.test(data7mu5$usia, data7mu5$material)

Levene's Test for Homogeneity of Variance
  Df F value Pr(>F)
group 2  1.0445  0.3632
 33
```

Seperti pada hasil **Uji Bartlett** sebelumnya, hasil **Uji Levene** ini menunjukkan bahwa tidak ada perbedaan varians dari usia pakai baterai pada ketiga jenis material yang digunakan dalam eksperimen. Hal ini ditunjukkan oleh *p-value* (yaitu 0.3632) yang lebih besar dari $\alpha=0.05$.

7.3. Pengujian Proporsi

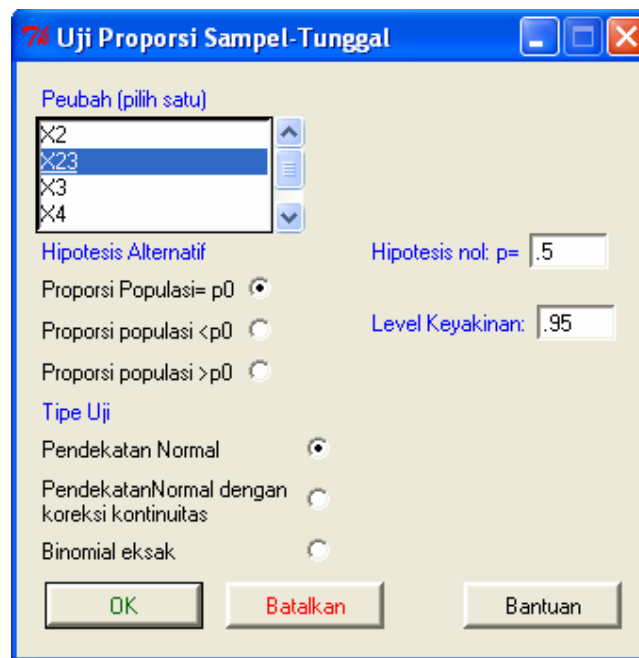
R menyediakan dua macam pilihan pada pengujian proporsi, yaitu **Uji Proporsi Sampel Tunggal** dan **Uji Proporsi Dua Sampel**. Pilihan-pilihan analisis statistika tersebut dapat diperoleh dengan memilih menu **Statistika**, dan kemudian memilih **Proporsi**. Untuk penjelasan pengujian proporsi ini digunakan data **HBAT.SAV** yang ada di buku Hair dkk. (2006, hal. 28-31) dengan judul **Multivariate Data Analysis**. Data tersebut berisi data profil responden, tingkat persepsi terhadap variabel-variabel pemasaran (kualitas produk, image website, kecepatan pengiriman, dan lain-lain), serta tingkat kepuasan konsumen.

Sebagai tahap awal, gunakan **R-Commander** untuk melakukan impor data file **SPSS** yaitu **HBAT.SAV** ke file **R**. Untuk itu, gunakan menu **Data**, pilih **Impor data**, dan kemudian klik **dari dataset SPSS....** Setelah itu, pengujian proporsi dengan **R** dapat dilakukan dengan pilihan-pilihan menu yang tersedia.

7.3.1. Pengujian Proporsi Sampel Tunggal

Pada data HBAAT ada salah satu pertanyaan yang berkaitan dengan apakah konsumen akan melakukan hubungan (memesan kembali) di masa yang akan datang dengan perusahaan (dinotasikan **X23**). Ada dua jawaban yang dapat dipilih, yaitu TIDAK dan YA. Misalkan dari 100 konsumen yang telah memberikan jawaban ingin diketahui apakah ada perbedaan proporsi yang menjawab TIDAK dan YA. Hal ini sama dengan pengujian untuk mengetahui apakah proporsi konsumen yang menjawab TIDAK (tidak mau melakukan hubungan di masa datang) adalah 0,50.

Uji proporsi sampel tunggal pada **R** disediakan melalui menu **Statistika**, pilih **Proporsi**, dan setelah itu pilih **Uji proporsi Sampel-tunggal...**, sehingga diperoleh jendela dialog seperti pada Gambar 7.18 berikut ini.



Gambar 7.18. Jendela dialog untuk Uji Proporsi Sampel-Tunggal

Selanjutnya, pilih **X23** pada jendela **Peubah**, dan isikan angka **0.5** pada jendela **Hipotesis nol: p=**. Seperti contoh sebelumnya, ada tiga pilihan **Hipotesis Alternatif** atau H_1 yaitu **Proporsi Populasi = p_0** , **Proporsi Populasi < p_0** , dan **Proporsi Populasi > p_0** . Pada contoh ini klik **Proporsi Populasi = p_0** sesuai dengan hipotesis statistika yang digunakan yaitu

$$H_0 : p = 0,50 \text{ atau proporsi yang menjawab TIDAK adalah } 0,50$$

$$H_1 : p \neq 0,50$$

Setelah itu tetapkan **Level Keyakinan** yang digunakan dalam pengujian (misalkan 0,95 yang berarti $\alpha=0.05$) beserta **Tipe Ujinya**. Klik **OK** sehingga diperoleh tampilan output pada jendela keluaran seperti berikut ini.

```
> hbat <- read.spss("D:/hair_multivariate_6_data/HBAT.sav",
  use.value.labels=TRUE, max.value.labels=Inf, to.data.frame=TRUE)

> .Table <- xtabs(~ X23 , data= hbat )

> .Table
  X23
  No, would not consider  Yes, would consider
                55                45

> prop.test(rbind(.Table), alternative='two.sided', p=.5, conf.level=.95,
  correct=FALSE)

1-sample proportions test without continuity correction

data:  rbind(.Table), null probability 0.5
X-squared = 1, df = 1, p-value = 0.3173
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.4524460 0.6438546
sample estimates:
 p
0.55
```

Dalam kasus ini, statistik uji yang digunakan adalah uji *Chi-square* atau χ^2 , yaitu (Johnson dan Bhattacharyya, 1996)

$$\chi^2 = \sum_{i=1}^2 \frac{(O_i - E_i)^2}{E_i},$$

dengan $E_i = np = 100 \times 0,5 = 50$ untuk masing-masing i .

Hasil output di atas menunjukkan bahwa pengujian gagal menolak H_0 yaitu proporsi konsumen yang menjawab TIDAK mau menjalin kembali hubungan di masa datang adalah 0,50. Hal ini ditunjukkan oleh *p-value* (yaitu 0.3173) yang lebih besar dari $\alpha=0.05$. Dengan demikian dapat disimpulkan bahwa tidak ada perbedaan proporsi konsumen yang mau dan tidak mau menjalin kembali hubungan dengan perusahaan di masa datang.

7.3.2. Pengujian Proporsi Dua Sampel

Salah satu variabel profile konsumen pada data HBAT adalah jenis perusahaan (dinotasikan **X2**), yaitu *magazine industry* dan *newsprint industry*. Misalkan ingin diketahui apakah ada perbedaan proporsi yang menjawab TIDAK dan YA pada pertanyaan tentang mau tidaknya melakukan hubungan kembali di masa datang dalam kedua kelompok konsumen industri tersebut.

Uji proporsi dua sampel adalah uji statistik yang dapat dilakukan untuk menjawab permasalahan tersebut. R menyediakan fasilitas uji ini melalui menu **Statistika**, pilih **Proporsi**, dan setelah itu pilih **Uji proporsi dua sampel...**, sehingga diperoleh jendela dialog seperti pada Gambar 7.19.



Gambar 7.19. Jendela dialog untuk **Uji Proporsi dua-sampel**

Selanjutnya, pilih **X2** pada jendela **Kelompok**, dan pilih **X23** pada jendela **Peubah respon**. Pada contoh ini hipotesis statistika yang digunakan adalah

$H_0 : p_1 = p_2$ atau proporsi yang menjawab TIDAK mau melakukan hubungan kembali dimasa datang pada konsumen *magazine industry* dan *newsprint industry* adalah SAMA

$H_1 : p_1 \neq p_2$ atau ada PERBEDAAN proporsi yang menjawab TIDAK mau melakukan hubungan kembali dimasa datang pada konsumen *magazine industry* dan *newsprint industry*

Setelah itu pilih **Hipotesis Alternatif** atau H_1 yang sesuai dengan hipotesis diatas, yaitu **Dua-arah**. Tetapkan juga **Level Keyakinan** dan **Tipe Uji** yang digunakan. Klik **OK** sehingga diperoleh tampilan output pada jendela keluaran seperti berikut ini.

```
> .Table <- xtabs(~X2+X23, data=hbat)

> .Table
```

	X23	
X2	No, would not consider	Yes, would consider
Magazine industry	30	22
Newsprint industry	25	23

```

> rowPercents(.Table)
      X23
X2      No, would not consider Yes, would consider Total Count
Magazine industry      57.7      42.3 100    52
Newsprint industry     52.1      47.9 100    48

> prop.test(.Table, alternative='two.sided', conf.level=.95, correct=FALSE)

      2-sample test for equality of proportions without continuity
      correction

data: .Table
X-squared = 0.3173, df = 1, p-value = 0.5732
alternative hypothesis: two.sided
95 percent confidence interval:
-0.1388571 0.2510366
sample estimates:
prop 1      prop 2
0.5769231 0.5208333

```

Pada contoh kasus ini, statistik uji yang digunakan adalah uji *Chi-square* atau χ^2 seperti pada sub-bab 6.8 sebelumnya, yaitu (Johnson dan Bhattacharyya, 1996)

$$\chi^2 = \sum_{i=1}^2 \sum_{j=1}^2 \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

dengan

O_{ij} = jumlah pengamatan pada baris ke- i dan kolom ke- j ,

E_{ij} = nilai ekspektasi pengamatan pada baris ke- i dan kolom ke- j .

Daerah penolakan untuk pengujian perbedaan kedua proporsi ini, yaitu tolak H_0 yang berarti ada perbedaan proporsi yang menjawab TIDAK mau melakukan hubungan kembali dimasa datang pada konsumen *magazine industry* dan *newsprint industry* adalah jika nilai

$$\chi^2 > \chi^2_{\alpha, df=1} \quad \text{atau} \quad \text{nilai } p < \alpha .$$

Hasil output di atas menunjukkan bahwa pengujian gagal menolak H_0 yaitu proporsi konsumen yang menjawab TIDAK mau menjalin kembali hubungan di masa datang antara konsumen *magazine industry* dan *newsprint industry* adalah SAMA. Hal ini ditunjukkan oleh *p-value* (yaitu 0.5732) yang lebih besar dari $\alpha=0.05$. Dengan demikian dapat disimpulkan bahwa jenis industri dari konsumen tidak memberikan perbedaan terhadap kemauan dalam menjalin kembali hubungan dengan perusahaan di masa datang.

BAB 8

ANALISIS REGRESI MENGGUNAKAN R-Commander

Ada berbagai prosedur dan *library* untuk melakukan analisis data dengan berbagai jenis persamaan regresi yang disediakan oleh **R**, baik model regresi linear ataupun nonlinear. Pada bagian ini akan dijelaskan analisis regresi linear yang disediakan di **R-Commander**. Secara umum ada dua menu yang disediakan **R-Commander** untuk analisis regresi linear, yaitu **Regresi Linear** dan **Model Linear**.

8.1. Regresi Linear

Secara umum bentuk matematis dari model regresi linier sederhana dapat dinyatakan sebagai berikut (Draper dan Smith, 1981; Kutner dkk., 2004)

$$\hat{Y} = \beta_0 + \beta_1 X$$

atau

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

dengan

- Y = nilai pengamatan dari peubah atau variabel tak bebas (respon)
- X = nilai pengamatan dari peubah atau variabel bebas (prediktor)
- \hat{Y} = nilai ramalan atau prediksi dari peubah atau variabel tak bebas (respon)
- ε = nilai kesalahan ramalan
- β_0 = intersep atau konstanta
- β_1 = *slope* atau koefisien kemiringan model regresi.

Metode kuadrat terkecil atau *ordinary least squares* (OLS) adalah suatu metode yang digunakan untuk menentukan suatu garis lurus atau menaksir nilai β_0 dan β_1 dengan kesesuaian terbaik yang meminimumkan jumlah kuadrat penyimpangan nilai Y yang diamati dari nilai-nilai yang diramalkan (jumlah kuadrat kesalahan atau $\sum \varepsilon^2$). Secara matematis metode ini adalah meminimumkan

$$\begin{aligned} \text{SSE} &= \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \\ &= \sum_{i=1}^n (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2. \end{aligned}$$

Dengan menggunakan differensial terhadap β_0 dan β_1 akan diperoleh nilai taksiran kuadrat terkecil untuk β_0 dan β_1 .

Berikut ini adalah rumus untuk mendapatkan nilai-nilai taksiran β_0 dan β_1 dengan menggunakan OLS, (Draper dan Smith, 1981)

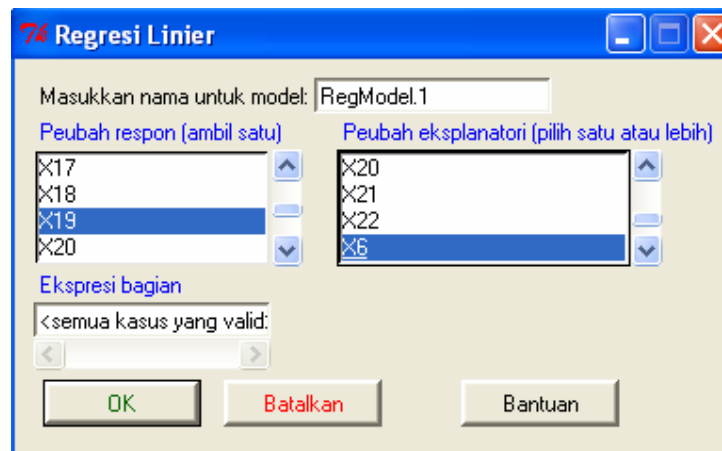
$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2}$$

$$= \frac{\sum_{i=1}^n X_i Y_i - n\bar{X}\bar{Y}}{\sum_{i=1}^n X_i^2 - n\bar{X}^2},$$

dan

$$\hat{\beta}_0 = \bar{Y} - \beta_1 \bar{X}.$$

Misalkan akan diamati hubungan antara tingkat persepsi konsumen terhadap kualitas produk HBAT (variabel **X6**) dan tingkat kepuasan konsumen (variabel **X19**) melalui model regresi linear. Untuk keperluan ini, **R-Commander** menyediakan fasilitas melalui menu **Statistika**, pilih **Pencocokan Model**, dan setelah itu pilih **Regresi Linier...**, sehingga diperoleh jendela dialog seperti pada Gambar 8.1. Menu ini disediakan terutama untuk estimasi model regresi linear dari variabel dependen yang bersifat metrik dengan variabel independen yang **semuanya** bersifat metrik, dan secara **default** memuat komponen konstanta dalam model regresinya.



Gambar 8.1. Jendela dialog untuk **Regresi Linier**

Pada contoh kasus HBAT ini, ketik nama untuk model regresi linear yang akan diestimasi (**default** adalah **Regmodel.1**). Hal ini berarti output hasil estimasi regresi linear disimpan sebagai objek dengan nama **Regmodel.1**. Kemudian pilih **X19** (tingkat kepuasan konsumen) pada jendela **Peubah respon** (variabel dependen), dan pilih **X6** (tingkat persepsi konsumen terhadap kualitas produk HBAT) pada jendela **Peubah eksplanatori** (variabel independen). Jendela dialog pada variabel independen menyediakan pilihan satu atau lebih yang mengindikasikan bahwa menu ini secara umum dapat digunakan untuk analisis regresi linear berganda. Klik **OK** sehingga diperoleh output model regresi linear sederhana seperti berikut ini.

```
> RegModel.1 <- lm(X19~X6, data=hbat)
> summary(RegModel.1)
```

Call:
lm(formula = X19 ~ X6, data = hbat)

Residuals:

Min	1Q	Median	3Q	Max
-1.88746	-0.72711	-0.01577	0.85641	2.25220

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.67593	0.59765	6.151	1.68e-08 ***
X6	0.41512	0.07534	5.510	2.90e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.047 on 98 degrees of freedom
Multiple R-Squared: 0.2365, Adjusted R-squared: 0.2287
F-statistic: 30.36 on 1 and 98 DF, p-value: 2.901e-07

```
> # perhatikan hasil dari perintah-perintah berikut ini
> hbat$fitted.RegModel.1 <- fitted(RegModel.1)
> hbat$residuals.RegModel.1 <- residuals(RegModel.1)
> hbat$rstudent.RegModel.1 <- rstudent(RegModel.1)
> hbat$hatvalues.RegModel.1 <- hatvalues(RegModel.1)
> hbat$cooks.distance.RegModel.1 <- cooks.distance(RegModel.1)

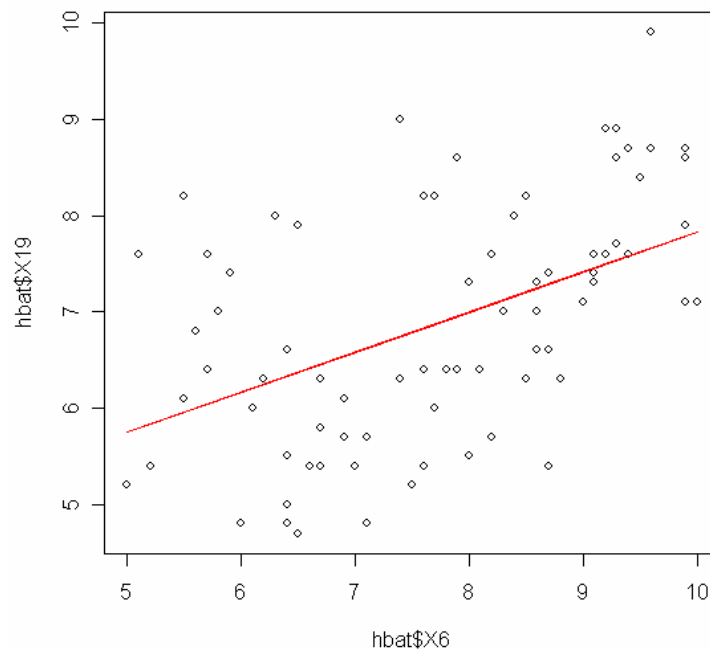
> # Perintah untuk mendapatkan gambar garis regresi
> plot(hbat$X6,hbat$X19)
> lines(hbat$X6, hbat$fitted.RegModel.1, col="red")
```

Output tersebut menunjukkan bahwa model linear regresi yang menunjukkan hubungan antara X yaitu persepsi kualitas produk terhadap Y yaitu kepuasan konsumen berdasarkan data sampel adalah

$$\hat{Y} = 3,67593 + 0,41512X$$

Model regresi linear ini menjelaskan bahwa ada pengaruh positif antara persepsi kualitas produk terhadap kepuasan konsumen. Nilai dugaan *slope* sebesar 0,41512 dapat diinterpretasikan sebagai kenaikan rata-rata tingkat kepuasan konsumen akibat kenaikan per satuan persepsi kualitas produk. Hasil uji t menunjukkan bahwa pengaruh persepsi kualitas produk tersebut adalah signifikan secara statistik pada $\alpha=0.05$. Hal ini ditunjukkan oleh besarnya p -value dari uji t (yaitu 2.90e-07) yang lebih kecil dari $\alpha=0.05$.

Perintah terakhir pada output di atas adalah untuk mendapatkan garis regresi yang menjelaskan hubungan antara persepsi terhadap kualitas produk dengan kepuasan konsumen. Hasil dari perintah ini dapat dilihat pada Gambar 8.2. Dari gambar ini dapat dijelaskan bahwa secara keseluruhan terdapat variasi observasi yang besar dari garis regresi yang ada. Hal ini juga ditunjukkan oleh nilai koefisien determinasi (R^2) model yang cukup kecil, yaitu 0,2365.



Gambar 8.2. Plot observasi dan hasil garis regresi linear

8.2. Model Linear

Menu pilihan **Model Linier** pada **R-Commander** bersifat lebih umum daripada menu **Regresi Linier** sebelumnya. Pada menu ini, variabel dependen dibatasi hanya untuk variabel yang bersifat metrik. Sedangkan untuk variabel independen, tidak terbatas hanya untuk variabel yang bersifat metrik, tetapi juga dapat yang bersifat nonmetrik atau bertipe kategori (yang dalam pengolahan data menggunakan variabel *dummy*).

Secara umum model regresi linear yang melibatkan lebih dari satu variabel bebas (prediktor) dikenal dengan model regresi linear berganda. Bentuk matematis dari model regresi linear berganda adalah (Draper dan Smith, 1981; Kutner dkk., 2004)

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p$$

dengan

\hat{Y} = nilai ramalan atau prediksi dari peubah atau variabel tak bebas (respon)

X_i = nilai pengamatan dari variabel bebas (prediktor), dengan $i = 1, 2, \dots, p$

β_0 = intersep atau konstanta

β_i = *slope* atau koefisien kemiringan model regresi, dengan $i = 1, 2, \dots, p$.

Estimasi terhadap parameter dalam model regresi linear berganda tersebut ($\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2$) diperlukan untuk mendapatkan model tersebut. Seperti pada model linear sederhana, hal ini dapat dilakukan dengan menggunakan metode kuadrat terkecil (OLS).

Nilai taksiran koefisien regresi berganda ini dapat pula diperoleh dengan cara pendekatan matrik yaitu, (Draper dan Smith, 1981; Kutner dkk., 2004)

$$\hat{\beta} = (X'X)^{-1} X'Y$$

dengan

$\hat{\beta}$ = matriks taksiran parameter (ukuran $p \times 1$, dengan p adalah jumlah parameter yang ditaksir)

X = matriks variabel bebas (ukuran $n \times p$)

Y = matriks variabel tak bebas (ukuran $n \times 1$).

Untuk $p=2$, maka contoh penjelasan matriks di atas dapat ditulis sebagai berikut

$$\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix}, \quad X = \begin{bmatrix} 1 & X_{11} & X_{12} \\ 1 & X_{21} & X_{22} \\ 1 & X_{31} & X_{32} \\ \dots & \dots & \dots \\ 1 & X_{n1} & X_{n2} \end{bmatrix}, \quad \text{dan} \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \dots \\ Y_n \end{bmatrix}.$$

Misalkan akan diteliti hubungan antara tipe konsumen berdasarkan lamanya menjadi konsumen (variabel **X1**) dan tingkat persepsi konsumen terhadap kualitas produk HBAT (variabel **X6**) terhadap tingkat kepuasan konsumen (variabel **X19**) melalui model regresi linear berganda. Dalam hal ini **X1** merupakan variabel nonmetrik yang terdiri dari 3 kategori, yaitu **kurang dari 1 tahun**, **1-5 tahun**, dan **lebih dari 5 tahun**. Untuk keperluan analisis regresi linear berganda ini, **R-Commander** menyediakan fasilitas melalui menu **Statistika**, pilih **Pencocokan Model**, dan setelah itu pilih **Model Linier...**, sehingga diperoleh jendela dialog seperti pada Gambar 8.3.



Gambar 8.3. Jendela dialog untuk **Model Linier**

Dalam kasus ini, model regresi linear berganda yang akan dicari bentuknya adalah

$$\hat{Y} = \beta_0 + \beta_1 X_{11} + \beta_2 X_{12} + \beta_3 X_6 .$$

Untuk penyelesaian kasus tersebut, ketik nama objek output model regresi linear yang akan diestimasi (misal **LinearModel.2**). Hal ini berarti output hasil estimasi regresi linear berganda disimpan sebagai objek dengan nama **LinearModel.2**. Kemudian pilih **X19** (tingkat kepuasan konsumen) pada jendela **Formula Model:** (variabel dependen), dan pilih **X6 + X1** (tingkat persepsi konsumen terhadap kualitas produk HBAT dan jenis konsumen) pada jendela kanan dari **Formula Model** (variabel independen). Jendela dialog pada **Formula Model** menyediakan banyak pilihan dari model linear ataupun model yang dilinearkan dengan transformasi tertentu. Klik **OK** sehingga diperoleh output model regresi linear berganda seperti berikut ini.

```

> LinearModel.2 <- lm(X19 ~ X6 + X1, data=hbat)
> summary(LinearModel.2)

Call:
lm(formula = X19 ~ X6 + X1, data = hbat)

Residuals:
    Min       1Q   Median       3Q      Max
-1.85973 -0.63250 -0.05293  0.54987  2.11380

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.8334    0.5706   6.718 1.31e-09 ***
X6              0.2665    0.0778   3.426 0.000903 ***
X1[T.1 to 5 years] 1.5511    0.1998   7.763 8.99e-12 ***
X1[T.Over 5 years] 1.3940    0.2557   5.452 3.87e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8157 on 96 degrees of freedom
Multiple R-Squared: 0.5458, Adjusted R-squared: 0.5316
F-statistic: 38.45 on 3 and 96 DF, p-value: < 2.2e-16

> # Perhatikan hasil dari perintah-perintah berikut ini
> hbat$fitted.LinearModel.2 <- fitted(LinearModel.2)
> hbat$residuals.LinearModel.2 <- residuals(LinearModel.2)
> hbat$rstudent.LinearModel.2 <- rstudent(LinearModel.2)
> hbat$hatvalues.LinearModel.2 <- hatvalues(LinearModel.2)
> hbat$cooks.distance.LinearModel.2 <- cooks.distance(LinearModel.2)
> trellis.device(theme="col.whitebg")
> plot(all.effects(LinearModel.2), ask=FALSE)

```

Output ini menunjukkan bahwa jenis (lama menjadi) konsumen dan persepsi terhadap kualitas produk berpengaruh signifikan terhadap kepuasan konsumen. Hal ini ditunjukkan oleh besarnya *p-value* dari uji *t* pada kedua variabel tersebut yang lebih kecil dari $\alpha=0.05$. Khusus untuk variabel jenis konsumen, ada dua koefisien regresi yang ditampilkan, yaitu pada konsumen **1-5 tahun** (sebesar 1,5511) dan **lebih dari 5 tahun** (sebesar 1,3940). Tanda koefisien regresi yang positif pada kedua variabel *dummy* tersebut menjelaskan bahwa konsumen lama (**1-5 tahun** dan **lebih dari 5 tahun**) memiliki tingkat kepuasan lebih tinggi dibanding konsumen baru (**kurang dari 1 tahun**). Sebagai tambahan, hasil regresi linear berganda memberikan nilai koefisien determinasi (R^2) model yang lebih besar dibanding hasil regresi linear sebelumnya, yaitu naik dari 0,2365 menjadi 0,5458.

Dengan demikian, model regresi linear berganda yang diperoleh dari data pada kasus di atas adalah

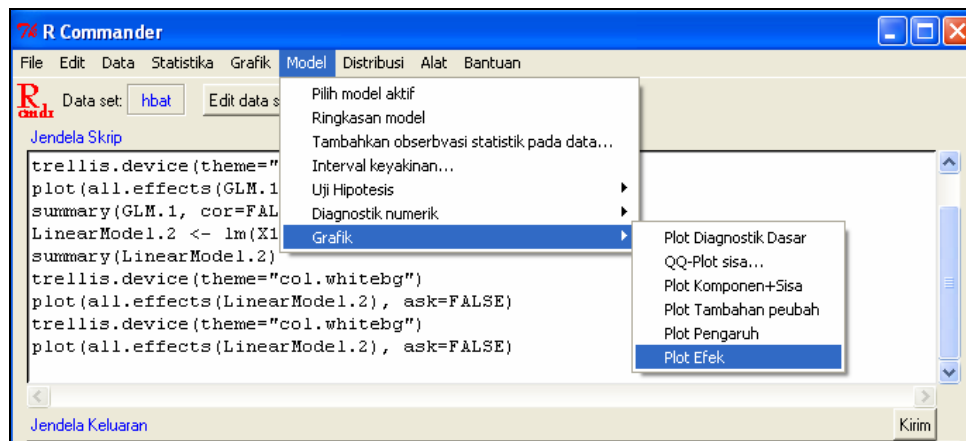
$$\hat{Y} = 3,8334 + 1,5511X_{1_1} + 1,3940X_{1_2} + 1,3940X_6$$

dengan X_{1_1} dan X_{1_2} adalah variabel *dummy*, yaitu

X_{1_1} bernilai 1 untuk konsumen **1-5 tahun**, dan NOL untuk konsumen yang lain,

X_{1_2} bernilai 1 untuk konsumen **lebih dari 5 tahun**, dan NOL untuk konsumen yang lain.

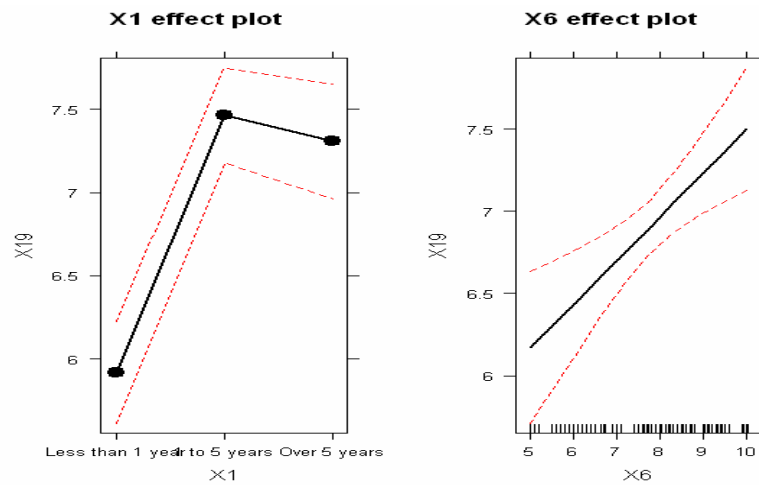
R-Commander juga menyediakan fasilitas analisis grafik tentang model linear regresi yang telah diperoleh, yaitu melalui menu **Model**, pilih **Grafik**, dan setelah itu pilih menu analisis grafik yang diinginkan. Menu pilihan-pilihan ini dapat diaktifkan setelah estimasi model linear sukses dijalankan. Berikut ini adalah menu pilihan analisis grafik yang disediakan oleh **R-Commander**.



Gambar 8.4. Jendela dialog untuk analisis lanjutan dari **Model Linier**

Klik pada pilihan **Plot Efek** akan menghasilkan output grafik seperti yang terlihat pada Gambar 8.5.

Dari gambar ini dapat dijelaskan bahwa tingkat kepuasan konsumen yang tertinggi ada pada konsumen **1-5 tahun**, sedangkan tingkat kepuasan yang terendah terletak pada konsumen baru yaitu **kurang dari 1 tahun**. Plot kedua menunjukkan bahwa variabel **X6** (tingkat persepsi kualitas produk) mempunyai pengaruh linear yang positif terhadap **X19**, yaitu tingkat kepuasan konsumen. Hal ini sesuai dengan tanda koefisien model regresi yang positif untuk **X6**.



Gambar 8.5. Plot Efek pada masing-masing variabel independen dari **Model Linier**

Berikut ini adalah ringkasan tentang prosedur uji serentak (simultan) dan uji individu (parsial) signifikansi koefisien pada model linear regresi.

- Uji **SERENTAK** atau **SIMULTAN** (Draper dan Smith, 1981; Kutner dkk., 2004)

1. Hipotesis pengujian

$$H_0 : \beta_1 = \beta_2 = \dots = \beta_p = 0$$

$$H_1 : \text{minimal ada satu } \beta_i \neq 0, i = 1, 2, \dots, p.$$

2. Statistik Uji

$$F = \frac{MS_{\text{Regresi}}}{MS_{\text{Error}}}$$

dengan

$$MS_{\text{Regresi}} = \frac{\sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2}{p-1}, \text{ dan } MS_{\text{Error}} = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n-p}.$$

3. Daerah penolakan

$$\text{Tolak } H_0 \text{ jika } F > F_{\alpha; (v1=p-1, v2=n-p)} \text{ atau } p\text{-value} < \alpha.$$

- Uji **INDIVIDU** atau **PARSIAL** (Draper dan Smith, 1981; Kutner dkk., 2004)

1. Hipotesis pengujian

$$H_0 : \beta_i = 0$$

$$H_1 : \beta_i \neq 0, i = 0, 1, 2, \dots, p.$$

2. Statistik Uji

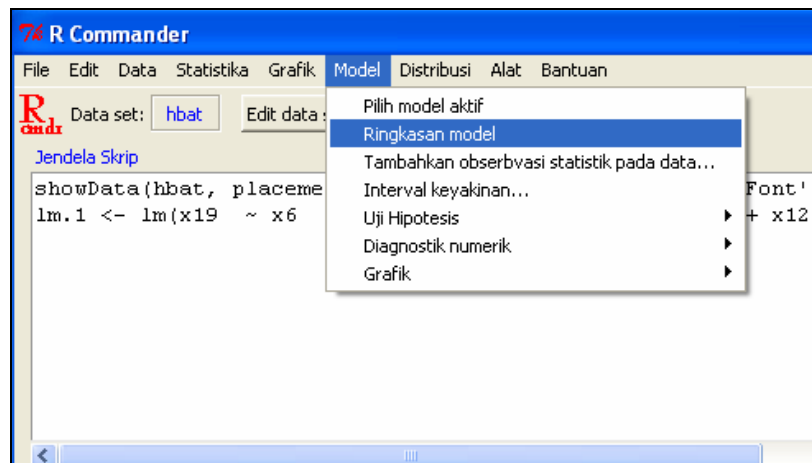
$$t = \frac{\hat{\beta}_i - 0}{\text{st.dev.}(\hat{\beta}_i)}.$$

3. Daerah penolakan

$$\text{Tolak } H_0 \text{ jika } |t| > t_{\frac{\alpha}{2}, df=n-p} \text{ atau } p\text{-value} < \alpha.$$

8.3. Cek Diagnosa Kesesuaian Model Regresi Linear

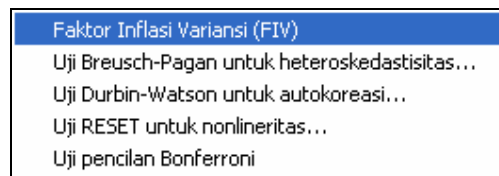
R-Commander menyediakan banyak fasilitas untuk evaluasi kesesuaian model regresi. Setelah suatu model regresi linear telah dijalankan dan diperoleh, maka semua pilihan pada menu **Model** aktif dan dapat dipilih untuk diaktifkan. Berikut ini adalah tampilan jendela pilihan pada pilihan menu **Model**.



Gambar 8.6. Jendela dialog untuk cek diagnosa dari suatu **Model Linier**

Jendela dialog untuk cek diagnosa suatu model regresi linear di atas menunjukkan bahwa ada banyak fasilitas yang disediakan **R-Commander** untuk evaluasi kesesuaian model regresi linear. Pada bagian ini, evaluasi kesesuaian model difokuskan pada deteksi multikolinearitas dan pengecekan kesesuaian asumsi model, yaitu $\varepsilon_i \sim IIDN(0, \sigma^2)$.

Besaran statistik yang biasanya digunakan untuk mendeteksi multikolinearitas antar variabel independen adalah *Variance-Inflation Factors* atau **VIF** (Kutner dkk., 2004). **R-Commander** menyediakan fasilitas untuk mengeluarkan besaran ini pada pilihan menu **Model**, dan kemudian **Diagnostik numerik**, sehingga muncul tampilan jendela dialog pilihan seperti berikut ini.



Gambar 8.7. Jendela dialog pilihan dari suatu **Diagnostik numerik**

Dari Gambar 8.7 dapat dilihat bahwa ada lima pilihan yang dapat dijalankan untuk evaluasi kesesuaian model, yaitu :

- deteksi multikolinearitas dengan besaran **VIF**,
- deteksi heteroskedastisitas dengan Uji **Breusch-Pagan**,
- deteksi autokorelasi dengan Uji **Durbin-Watson**,
- deteksi nonlinearitas dengan Uji **RESET**, dan
- deteksi pencilan dengan Uji **Bonferroni**.

Selain itu, **R-Commander** juga menyediakan fasilitas grafik untuk evaluasi kesesuaian asumsi model regresi linear. Residual dari model regresi merupakan besaran standar yang digunakan untuk evaluasi kesesuaian asumsi model regresi. Beberapa besaran lain yang juga banyak dipakai untuk evaluasi kesesuaian model regresi adalah *standardized residuals*, yaitu (Kutner dkk., 2004)

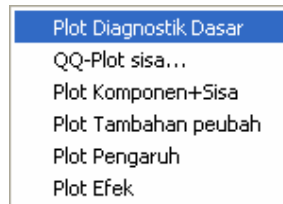
$$\varepsilon'_i = \frac{\varepsilon_i}{s\sqrt{1-h_{ii}}}$$

dengan s adalah akar dari *mean square error* (MSE), h_{ii} adalah diagonal dari matriks *hat* atau H yaitu

$$H = X(X^T X)^{-1} X^T.$$

h_{ii} adalah besaran yang dapat digunakan untuk mendeteksi pengaruh observasi Y_i .

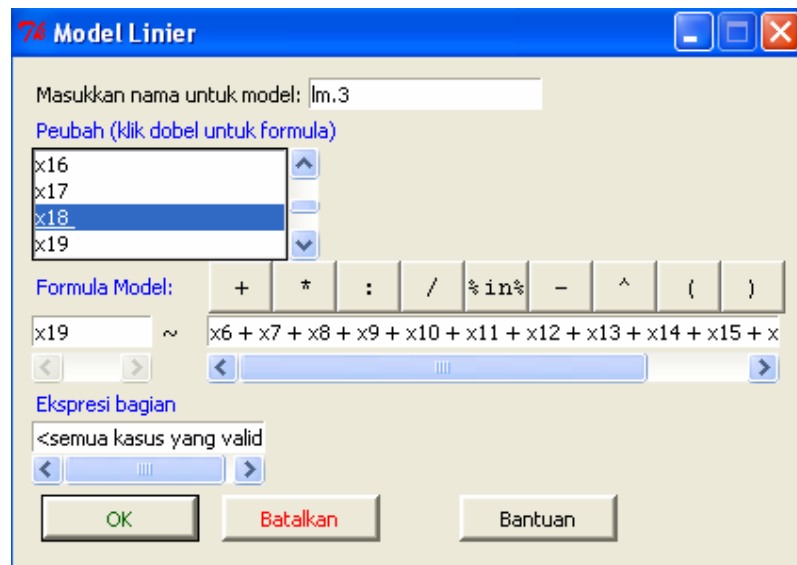
Fasilitas pada **R-Commander** untuk evaluasi kesesuaian model regresi dengan analisis grafik tersedia pada pilihan menu **Model**, dan kemudian **Grafik**, sehingga muncul tampilan jendela dialog pilihan seperti berikut ini.



Gambar 8.8. Jendela dialog pilihan evaluasi kesesuaian model dengan **Grafik**

Pilihan pada Gambar 8.8 menunjukkan bahwa ada enam grafik yang dapat dijalankan untuk evaluasi kesesuaian model, yaitu **Plot Diagnostik Dasar**, **QQ-Plot sisa**, **Plot Komponen+Sisa**, **Plot Tambahan peubah**, **Plot Pengaruh**, dan **Plot Efek**.

Misalkan akan diteliti hubungan antara tingkat persepsi konsumen terhadap beberapa variabel pemasaran dari produk HBAT (variabel **x6**, **x7**, ..., **x18**) dengan tingkat kepuasan konsumen (variabel **x19**) melalui model regresi linear berganda. Untuk itu jalankan kembali analisis regresi linear berganda dengan memasukkan variabel independen **x6**, **x7**, ..., **x18**, seperti yang terlihat pada jendela dialog di Gambar 8.9.



Gambar 8.9. Jendela dialog pilihan untuk **Model Regresi Linear Berganda**

Setelah semua isian pilihan lengkap, klik **OK** sehingga diperoleh output model regresi linear berganda pada jendela keluaran seperti berikut ini.

```
> lm.3 <- lm(x19 ~ x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13 + x14
+ x15 + x16 + x17 + x18, data=hbat)

> summary(lm.3)

Call:
lm(formula = x19 ~ x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13 +
x14 + x15 + x16 + x17 + x18, data = hbat)

Residuals:
    Min       1Q   Median       3Q      Max
-1.38704 -0.31208  0.08356  0.40652  0.91947

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.335836   1.120309  -1.192  0.23639
x6            0.377422   0.052707   7.161 2.55e-10 ***
x7           -0.456065   0.136509  -3.341  0.00124 **
x8            0.035203   0.064924   0.542  0.58907
x9            0.154286   0.103602   1.489  0.14009
x10          -0.034414   0.062829  -0.548  0.58529
x11           0.362389   0.266690   1.359  0.17775
x12           0.827376   0.101460   8.155 2.57e-12 ***
x13          -0.047465   0.048202  -0.985  0.32753
x14          -0.106968   0.125531  -0.852  0.39652
x15          -0.002939   0.039535  -0.074  0.94091
x16           0.143065   0.104519   1.369  0.17463
x17           0.237926   0.272485   0.873  0.38500
x18          -0.249168   0.514102  -0.485  0.62914
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

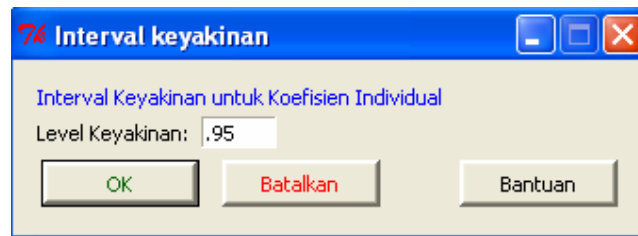
Residual standard error: 0.5663 on 86 degrees of freedom
Multiple R-squared: 0.8039, Adjusted R-squared: 0.7742
F-statistic: 27.11 on 13 and 86 DF, p-value: < 2.2e-16
```

Berdasarkan output ini, maka model regresi linear berganda yang diperoleh dari data pada kasus HBAT di atas adalah (tampilan dua angka di belakang koma)

$$\hat{Y} = -1,34 + 0,38X_6 - 0,46X_7 + 0,04X_8 + \dots + 0,24X_{17} - 0,25X_{18}.$$

Output di atas juga menunjukkan bahwa hanya ada tiga variabel independen yang berpengaruh signifikan terhadap kepuasan konsumen, yaitu X_6 , X_7 , dan X_{12} .

R-Commander juga menyediakan fasilitas untuk evaluasi signifikansi parameter model regresi dengan menggunakan interval keyakinan. Hal ini dapat dilakukan melalui menu **Model**, dan pilih **Interval keyakinan...**, sehingga diperoleh jendela dialog seperti berikut ini.



Gambar 8.10. Jendela dialog untuk **Interval keyakinan** koefisien model regresi

Klik **OK** sehingga diperoleh output model regresi linear berganda pada jendela keluaran seperti berikut ini.

```
> Confint(lm.3, level=.95)
```

	2.5 %	97.5 %
(Intercept)	-3.56293670	0.89126370
x6	0.27264462	0.48219908
x7	-0.72743631	-0.18469316
x8	-0.09386134	0.16426636
x9	-0.05166841	0.36023963
x10	-0.15931399	0.09048646
x11	-0.16777421	0.89255200
x12	0.62568021	1.02907262
x13	-0.14328849	0.04835827
x14	-0.35651558	0.14258055
x15	-0.08153160	0.07565348
x16	-0.06471161	0.35084127
x17	-0.30375712	0.77960892
x18	-1.27116865	0.77283249

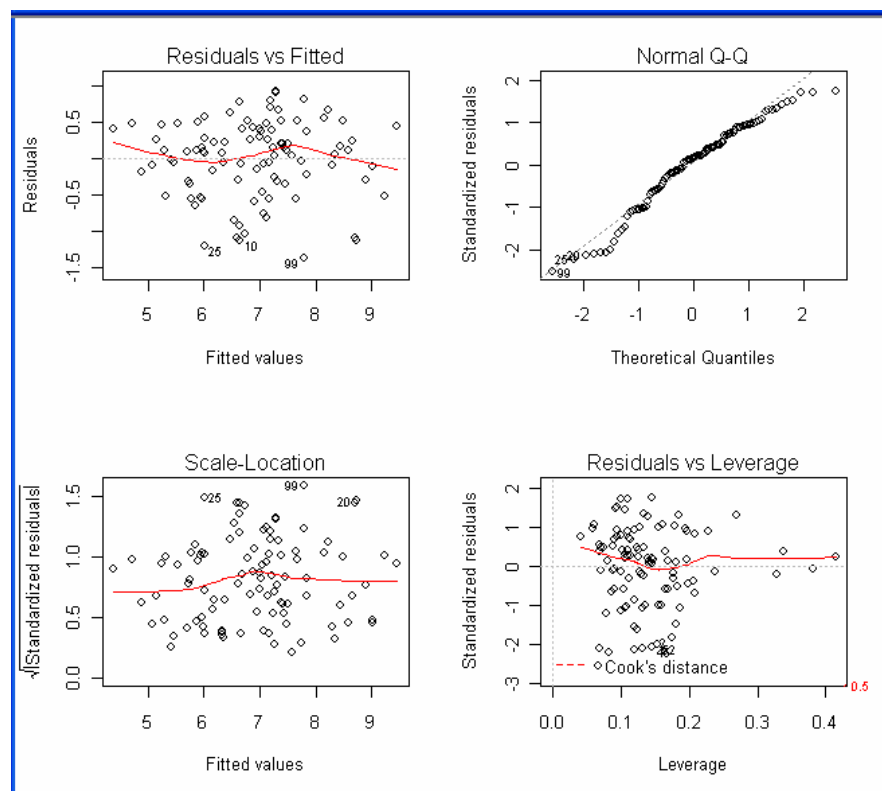
Selanjutnya, evaluasi kesesuaian model akan dilakukan untuk mendeteksi multikolinearitas antar variabel independen, yaitu melalui menu **Model**, pilih **Diagnostik numerik**, dan kemudian klik **Faktor Inflasi Variansi (FIV)** sehingga diperoleh output pada jendela keluaran seperti berikut ini.

```
> vif(lm.3)
```

x6	x7	x8	x9	x10	x11	x12
1.671693	2.822562	3.047401	4.837740	1.547421	37.978425	3.653611
x13	x14	x15	x16	x17	x18	
1.712022	3.268413	1.075445	2.909058	33.332337	44.003758	

Output ini menjelaskan bahwa variabel independen yang mempunyai kolinearitas tinggi adalah **x11**, **x17**, dan **x18**. Hal ini diindikasikan dengan nilai **VIF** yang besar yaitu lebih dari 10 (Kutner dkk., 2004)

Evaluasi selanjutnya adalah analisis grafik untuk melihat apakah residual sudah memenuhi syarat kesesuaian model. Hal ini dapat dilakukan melalui menu **Model**, pilih **Grafik**, dan kemudian klik **Plot Diagnostik Dasar** sehingga diperoleh output grafik seperti pada Gambar 8.11 berikut ini.

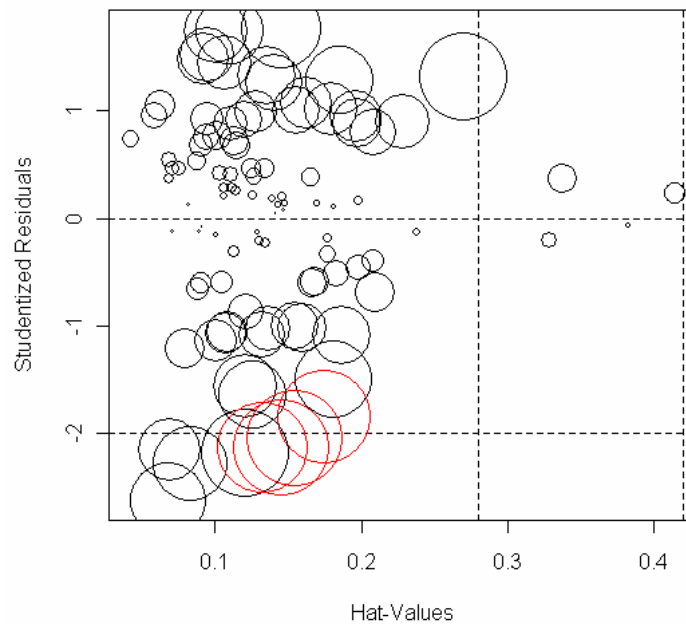


Gambar 8.11. Output **Plot Diagnostik Dasar** untuk evaluasi kesesuaian model

Output grafik di atas terdiri dari empat plot utama untuk evaluasi kesesuaian residual model, yaitu :

- Plot antara *fitted values* (nilai-nilai prediksi) dengan residual,
- Plot kuantil-kuantil normal dari *standardized residuals*,
- Plot antara *fitted values* dengan akar *standardized residuals*, dan
- Plot antara *leverage* dengan *standardized residuals*.

Selain itu, **R-Commander** juga memberikan fasilitas untuk evaluasi untuk deteksi adanya pengamatan yang berpengaruh ataupun pencilan dengan menampilkan plot antara nilai-nilai *hat* dengan *Studentized Residuals* (lihat Venables dan Ripley (1997), halaman 204-205). Hal ini dapat dilakukan melalui menu **Model**, pilih **Grafik**, dan kemudian klik **Plot Pengaruh** sehingga diperoleh output grafik seperti pada Gambar 8.12 berikut ini.



Gambar 8.12. Output **Plot Pengaruh** untuk deteksi *influence* dan pencilan

8.4. Rangkuman perintah dan library yang berkaitan dengan Analisis Regresi

Berikut ini adalah rangkuman **perintah** dan penjelasan tentang kegunaan, serta **library** dari **perintah** tersebut, yang biasanya digunakan dalam analisis regresi.

▪ Linear Model

Perintah	Kegunaan	library
Anova	Tabel Anova untuk model linear dan model linear tergeneralisir (GLM)	car
anova	Menghitung suatu tabel analisis varians untuk satu atau lebih model linear yang diestimasi	stats
coef	Suatu fungsi generic untuk mengekstraksi koefisien model dari obyek pada fungsi pemodelan	stats
coeftest	Pengujian koefisien yang diestimasi	lmtest
confint	Menghitung taksiran interval untuk satu atau lebih parameter dalam model yang diestimasi	stats
deviance	Mengembalikan deviance dari objek model yang diestimasi	stats
effects	Mengembalikan efek dari model yang diestimasi	stats
fitted	Suatu fungsi generic untuk mengekstraksi nilai prediksi objek yang diambil dari fungsi pemodelan	stats
formula	Memberikan suatu cara mengekstraksi formula yang terintegrasi dengan objek lain	stats
linear.hypothesis	Menguji hipotesis kelinearan	car
lm	Digunakan untuk mengestimasi model linear. Perintah ini dapat digunakan untuk regresi, analisis varians, dan analisis kovarians	stats
model.matrix	Membuat suatu matriks rancangan	stats
predict	Nilai prediksi berdasarkan objek model linear	stats
residuals	Suatu fungsi generic untuk mengekstraksi residual model dari objek pada fungsi pemodelan	stats
summary.lm	Metode untuk meringkas (summary)	stats
vcov	Mengambil matriks varians-kovarians dari parameter utama objek model yang diestimasi	stats

▪ **Pemilihan variabel dalam model**

Perintah	Kegunaan	library
add1	Menghitung semua argumen <i>'single terms'</i> yang ditambahkan atau dikeluarkan dari model, mengestimasi model dan menghitung tabel dari perubahan di nilai prediksi	stats
AIC	Fungsi <i>generic</i> untuk menghitung AIC atau <i>'Akaike information criterion'</i> untuk satu atau beberapa objek model yang diestimasi, berdasarkan rumus $-2 \cdot \log\text{-likelihood} + k \cdot \text{npar}$, dengan <i>'npar'</i> adalah jumlah parameter model, dan $k = 2$ untuk AIC, atau $k = \log(n)$ untuk BIC atau SBC (<i>Schwarz's Bayesian criterion</i>), dengan n adalah banyaknya pengamatan.	stats
Cpplot	Plot C_p	faraway
drop1	Menghitung semua argumen <i>'single terms'</i> yang ditambahkan atau dikeluarkan dari model, mengestimasi model dan menghitung tabel dari perubahan di nilai prediksi	stats
extractAIC	Menghitung (<i>generalized</i>) AIC untuk model parametrik yang diestimasi	stats
leaps	Pemilihan <i>'subset'</i> dengan <i>'leaps and bounds'</i>	leaps
maxadjr	Maximum Adjusted R-squared	faraway
offset	<i>An offset is a term to be added to a linear predictor, such as in a generalised linear model, with known coefficient 1 rather than an estimated coefficient</i>	stats
step	<i>Select a formula-based model by AIC</i>	stats
update.formula	<i>is used to update model formulae. This typically involves adding or dropping terms, but updates can be more general</i>	stats

▪ Cek diagnosa kesesuaian model

Perintah	Kegunaan	library
cookd	<i>Cook's Distances for Linear and Generalized Linear Models</i>	car
cooks.distance	<i>Cook's distance</i>	stats
covratio	<i>covariance ratio</i>	stats
dfbeta	DBETA	stats
dfbetas	DBETAS	stats
dffits	DFFITS	stats
hat	<i>diagonal elements of the hat matrix</i>	stats
hatvalues	<i>diagonal elements of the hat matrix</i>	stats
influence.measures	<i>This suite of functions can be used to compute some of theregression (leave-one-out deletion) diagnostics for linear and generalized linearmodels</i>	stats
lm.influence	<i>This function provides the basic quantities which are used in forminga wide variety of diagnostics for checking the quality of regression fits</i>	stats
ls.diag	<i>Computes basic statistics, including standard errors, t-and p-values forthe regression coefficients</i>	stats
outlier.test	<i>Bonferroni Outlier Test</i>	car
rstandard	<i>standardized residuals</i>	stats
rstudent	<i>studentized residuals</i>	stats
vif	<i>Variance Inflation Factor</i>	car

▪ Analisis Grafik pada model linear

Perintah	Kegunaan	library
ceres.plots	<i>Ceres Plots</i>	car
cr.plots	<i>Component+Residual (Partial Residual) Plots</i>	car
influence.plot	<i>Regression Influence Plot</i>	car
leverage.plots	<i>Regression Leverage Plots</i>	car
panel.car	<i>Panel Function Coplots</i>	car
plot.lm	<i>Four plots (selectable by which) are currently provided: a plot of residuals against fitted values, a Scale-Location plot of $\sqrt{ residuals }$ against fitted values, a Normal Q-Q plot, and a plot of Cook's distances versus row labels</i>	stats
prplot	<i>Partial Residual Plot</i>	faraway
qq.plot	<i>Quantile-Comparison Plots</i>	car
qqline	<i>adds a line to a normal quantile-quantile plot which passes through the first and third quartiles</i>	stats
qqnorm	<i>is a generic function the default method of which produces a normal QQ plot of the values in y</i>	stats
reg.line	<i>Plot Regression Line</i>	car
scatterplot.matrix	<i>Scatterplot Matrices</i>	car
scatterplot	<i>Scatterplots with Boxplots</i>	car
spread.level.plot	<i>Spread-Level Plots</i>	car

▪ Pengujian asumsi pada model linear

Perintah	Kegunaan	library
ad.test	<i>Anderson-Darling test for normality</i>	nortest
bartlett.test	<i>Performs Bartlett's test of the null that the variances in each of the groups (samples) are the same</i>	stats
bgtest	<i>Breusch-Godfrey Test</i>	lmtest
bptest	<i>Breusch-Pagan Test</i>	lmtest
cvm.test	<i>Cramer-von Mises test for normality</i>	nortest
durbin.watson	<i>Durbin-Watson Test for Autocorrelated Errors</i>	car
dwtest	<i>Durbin-Watson Test</i>	lmtest
levene.test	<i>Levene's Test</i>	car
lillie.test	<i>Lilliefors (Kolmogorov-Smirnov) test for normality</i>	nortest
ncv.test	<i>Score Test for Non-Constant Error Variance</i>	car
pearson.test	<i>Pearson chi-square test for normality</i>	nortest
sf.test	<i>Shapiro-Francia test for normality</i>	nortest
shapiro.test	<i>Performs the Shapiro-Wilk test of normality</i>	stats

▪ Transformasi Variabel pada model linear

Perintah	Kegunaan	library
box.cox	<i>Box-Cox Family of Transformations</i>	car
boxcox	<i>Box-Cox Transformations for Linear Models</i>	MASS
box.cox.powers	<i>Multivariate Unconditional Box-Cox Transformations</i>	car
box.tidwell	<i>Box-Tidwell Transformations</i>	car
box.cox.var	<i>Constructed Variable for Box-Cox Transformation</i>	car

▪ **Regresi Ridge**

Perintah	Kegunaan	library
<code>lm.ridge</code>	<i>Ridge Regression</i>	MASS

▪ **Regresi tersegmentasi (Segmented Regression)**

Perintah	Kegunaan	library
<code>segmented</code>	<i>Segmented relationships in regression models</i>	segmented
<code>slope.segmented</code>	<i>Summary for slopes of segmented relationships</i>	segmented

▪ **Least Squares Tergeneralisir (Generalized Least Squares)**

Perintah	Kegunaan	library
<code>ACF.gls</code>	<i>Autocorrelation Function for gls Residuals</i>	nlme
<code>anova.gls</code>	<i>Compare Likelihoods of Fitted Objects</i>	nlme
<code>gl</code>	<i>Fit Linear Model Using Generalized Least Squares</i>	nlme
<code>intervals.gls</code>	<i>Confidence Intervals on gls Parameters</i>	nlme
<code>lm.gls</code>	<i>fit Linear Models by Generalized Least Squares</i>	MASS
<code>plot.gls</code>	<i>Plot a gls Object</i>	nlme
<code>predict.gls</code>	<i>Predictions from a gls Object</i>	nlme
<code>qqnorm.gls</code>	<i>Normal Plot of Residuals from a gls Object</i>	nlme
<code>residuals.gls</code>	<i>Extract gls Residuals</i>	nlme
<code>summary.gls</code>	<i>Summarize a gls Object</i>	nlme

▪ **Model Linear Tergeneralisir (Generalized Linear Model)**

Perintah	Kegunaan	library
family	<i>Family objects provide a convenient way to specify the details of the models used by functions such as glm</i>	stats
glm.nb	<i>fit a Negative Binomial Generalized Linear Model</i>	MASS
glm	<i>is used to fit generalized linear models, specified by giving a symbolic description of the linear predictor and a description of the error distribution</i>	stats
polr	<i>Proportional Odds Logistic Regression</i>	MASS

▪ **Least Squares Nonlinear (Nonlinear Least Squares atau NLS)**

Perintah	Kegunaan	library
nlm	<i>This function carries out a minimization of the function f using a Newton-type algorithm</i>	stats
nls	<i>Determine the nonlinear least-squares estimates of the nonlinear model parameters and return a class nls object</i>	stats
nlscontrol	<i>Allow the user to set some characteristics of the nls nonlinear leastsquares algorithm</i>	stats
nlsModel	<i>This is the constructor for nlsModel objects, which are function closures for several functions in a list. The closure includes a nonlinear model formula, data values for the formula, as well as parameters and their values</i>	stats

▪ **Generalized Nonlinear Least Squares atau GNLS**

Perintah	Kegunaan	library
coef.gnls	<i>Extract gnls Coefficients</i>	nlme
gnls	<i>Fit Nonlinear Model Using Generalized Least Squares</i>	nlme
predict.gnls	<i>Predictions from a gnls Object</i>	nlme

▪ Loess Regression

Perintah	Kegunaan	library
loess	<i>Fit a polynomial surface determined by one or more numerical predictors, using local fitting</i>	stats
loess.control	<i>Set control parameters for loessfits</i>	stats
predict.loess	<i>Predictions from a loessfit, optionally with standard errors</i>	stats
scatter.smooth	<i>Plot and add a smooth curve computed by loess to a scatter plot</i>	stats

▪ Splines Regression

Perintah	Kegunaan	library
bs	<i>B-Spline Basis for Polynomial Splines</i>	splines
ns	<i>Generate a Basis Matrix for Natural C B-Spline Basis for Polynomial Splines</i>	splines
periodicSpline	<i>Create a Periodic Interpolation Spline</i>	splines
polySpline	<i>Piecewise Polynomial Spline Representation</i>	splines
predict.bSpline	<i>Evaluate a Spline at New Values of x</i>	splines
predict.bs	<i>Evaluate a Spline Basis</i>	splines
splineDesign	<i>Design Matrix for B-splines</i>	splines
splineKnots	<i>Knot Vector from a Spline</i>	splines
splineOrder	<i>Determine the Order of a Spline</i>	splines

▪ Robust Regression

Perintah	Kegunaan	library
lqs	<i>Resistant Regression</i>	MASS
rlm	<i>Robust Fitting of Linear Models</i>	MASS

▪ **Structural equation models**

Perintah	Kegunaan	library
sem	<i>General Structural Equation Models</i>	sem
tsls	<i>Two-Stage Least Squares</i>	sem

▪ **Simultaneous Equation Estimation**

Perintah	Kegunaan	library
systemfit	<i>Fits a set of linear structural equations using Ordinary Least Squares (OLS), Weighted Least Squares (WLS), Seemingly Unrelated Regression (SUR), Two-Stage Least Squares (2SLS), Weighted Two-Stage Least Squares (W2SLS) or Three-Stage Least Squares (3SLS)</i>	systemfit

▪ **Partial Least Squares Regression (PLSR) dan Principal Component Regression (PCR)**

Perintah	Kegunaan	library
biplot.mvr	<i>Biplots of PLSR and PCR Models</i>	pls
coefplot	<i>Plot Regression Coefficients of PLSR and PCR models</i>	pls
crossval	<i>Cross-validation of PLSR and PCR models</i>	pls
cvsegments	<i>Generate segments for cross-validation</i>	pls
kernelpls.fit	<i>Kernel PLS (Dayal and MacGregor)</i>	pls
msc	<i>Multiplicative Scatter Correction</i>	pls
mvr	<i>Partial Least Squares and Principal Components Regression</i>	pls
mvrCv	<i>Cross-validation</i>	pls
oscorespls.fit	<i>Orthogonal scores PLSR</i>	pls
predplot	<i>Prediction Plots</i>	pls
scoreplot	<i>Plots of Scores and Loadings</i>	pls

Lanjutan: PLSR and PCR

Perintah	Kegunaan	library
scores	<i>Extract Scores and Loadings from PLSR and PCR Models</i>	pls
svdpc.fit	<i>Principal Components Regression</i>	pls
validationplot	<i>Validation Plots</i>	pls

▪ **Quantile Regression**

Perintah	Kegunaan	library
anova.rq	<i>Anova function for quantile regression fits</i>	quantreg
boot.rq	<i>Bootstrapping Quantile Regression</i>	quantreg
lprq	<i>locally polynomial quantile regression</i>	quantreg
nlrq	<i>Function to compute nonlinear quantile regression estimates</i>	quantreg
qss	<i>Additive Nonparametric Terms for rqss Fitting</i>	quantreg
ranks	<i>Quantile Regression Ranks</i>	quantreg
rq	<i>Quantile Regression</i>	quantreg
rqss	<i>Additive Quantile Regression Smoothing</i>	quantreg
rrs.test	<i>Quantile Regression Rankscore Test</i>	quantreg
standardize	<i>Function to standardize the quantile regression process</i>	quantreg

▪ Linear and nonlinear mixed effects models

Perintah	Kegunaan	library
ACF	<i>Autocorrelation Function</i>	nlme
ACF.lme	<i>Autocorrelation Function for lme Residuals</i>	nlme
anova.lme	<i>Compare Likelihoods of Fitted Objects</i>	nlme
fitted.lme	<i>Extract lme Fitted Values</i>	nlme
fixed.effects	<i>Extract lme Fitted Values</i>	nlme
intervals	<i>Confidence Intervals on Coefficients</i>	nlme
intervals.lme	<i>Confidence Intervals on lme Parameters</i>	nlme
lme	<i>Linear Mixed-Effects Models</i>	nlme
nlme	<i>Nonlinear Mixed-Effects Models</i>	nlme
predict.lme	<i>Predictions from an lme Object</i>	nlme
predict.nlme	<i>Predictions from an nlme Object</i>	nlme
qqnorm.lme	<i>Normal Plot of Residuals or Random Effects from an lme object</i>	nlme
random.effects	<i>Extract Random Effects</i>	nlme
ranef.lme	<i>Extract lme Random Effects</i>	nlme
residuals.lme	<i>Extract lme Residuals</i>	nlme
simulate.lme	<i>Simulate lme models</i>	nlme
summary.lme	<i>Summarize an lme Object</i>	nlme
glmmPQL	<i>Fit Generalized Linear Mixed Models via PQL</i>	MASS

▪ **Generalized Additive Model (GAM)**

Perintah	Kegunaan	library
anova.gam	<i>Compare the fits of a number of gam models</i>	gam
gam.control	<i>Control parameters for fitting gam models</i>	gam
gam	<i>Fit a generalized additive model</i>	gam
na.gam.replace	<i>A missing value method that is helpful with gam</i>	gam
plot.gam	<i>An interactive plotting function for gam</i>	gam
predict.gam	<i>Make predictions from a gam object</i>	gam
preplot.gam	<i>extracts the components from a gam in a plot-ready form</i>	gam
step.gam	<i>stepwise model search with gam</i>	gam
summary.gam	<i>summary method for gam</i>	gam

▪ **Survival Analysis**

Perintah	Kegunaan	library
anova.survreg	<i>ANOVA tables for survreg objects</i>	survival
clogit	<i>Conditional logistic regression</i>	survival
cox.zph	<i>Test the proportional hazards assumption of a Cox regression</i>	survival
coxph	<i>Proportional Hazards Regression</i>	survival
oxph.detail	<i>Details of a cox model fit</i>	survival
coxph.rvar	<i>Robust variance for a Cox model</i>	survival
ridge	<i>Ridge regression</i>	survival
survdiff	<i>Test Survival Curve Differences</i>	survival
survexp	<i>Compute Expected Survival</i>	survival
survfit	<i>Compute a survival Curve for Censored Data</i>	survival
survreg	<i>Regression for a parametric survival model</i>	survival

▪ **Classification and Regression Trees**

Perintah	Kegunaan	library
cv.tree	<i>Cross-validation for Choosing tree Complexity</i>	tree
deviance.tree	<i>Extract Deviance from a tree Object</i>	tree
labels.rpart	<i>Create Split Labels For an rpart Object</i>	rpart
meanvar.rpart	<i>Mean-Variance Plot for an rpart Object</i>	rpart
misclass.tree	<i>Misclassifications by a Classification tree</i>	tree
na.rpart	<i>Handles Missing Values in an rpart Object</i>	rpart
partition.tree	<i>Plot the Partitions of a simple Tree Model</i>	tree
path.rpart	<i>Follow Paths to Selected Nodes of an rpart Object</i>	rpart
plotcp	<i>Plot a Complexity Parameter Table for an rpart Fit</i>	rpart
printcp	<i>Displays CP table for Fitted rpart Object</i>	rpart
prune.misclass	<i>Cost-complexity Pruning of Tree by error rate</i>	tree
prune.rpart	<i>Cost-complexity Pruning of an rpart Object</i>	rpart
prune.tree	<i>Cost-complexity Pruning of tree Object</i>	tree
rpart	<i>Recursive Partitioning and Regression Trees</i>	rpart
rpconvert	<i>Update an rpart object</i>	rpart
rsq.rpart	<i>Plots the Approximate R-Square for the Different Splits</i>	rpart
snip.rpart	<i>Snip Subtrees of an rpart Object</i>	rpart
solder	<i>Soldering of Components on Printed-Circuit Boards</i>	rpart
text.tree	<i>Annotate a Tree Plot</i>	tree
tile.tree	<i>Add Class Barplots to a Classification Tree Plo</i>	tree
tree.control	<i>Select Parameters for Tree</i>	tree
tree.screens	<i>Split Screen for Plotting Trees</i>	tree
tree	<i>Fit a Classification or Regression Tree</i>	tree

▪ **Beta regression**

Perintah	Kegunaan	library
betareg	<i>Fitting beta regression models</i>	betareg
plot.betareg	<i>Plot Diagnostics for a betareg Object</i>	betareg
predict.betareg	<i>Predicted values from beta regression model</i>	betareg
residuals.betareg	<i>Residuals function for beta regression models</i>	betareg
summary.betareg	<i>Summary method for Beta Regression</i>	betareg

BAB 9

GENERALIZED LINEAR MODEL MENGGUNAKAN R

Model Linier Tergeneralisir atau *Generalized Linear Model* (GLM) merupakan pengembangan dari model linear yang mengakomodir dua hal utama, yaitu distribusi respon yang non-normal dan transformasi untuk linearitas. Referensi yang komprehensif tentang GLM dapat dilihat di McCullagh dan Nelder (1989). Pada bab ini akan dibahas pengantar teori Model Linear Tergeneralisir (GLM) dan contoh suatu kasus GLM dengan menggunakan **R-Commander**.

9.1. Pengantar Teori Model Linear Tergeneralisir

Suatu Model Linear Tergeneralisir dapat dideskripsikan oleh asumsi-asumsi berikut ini :

- Ada suatu respon, y , teramati secara independen pada nilai-nilai yang tetap dari variabel-variabel *stimulus* x_1, x_2, \dots, x_p .
- Variabel-variabel *stimulus* hanya mempengaruhi distribusi dari y melalui suatu fungsi linear tunggal yang disebut dengan prediktor linear $\eta = \beta_1 x_1 + \dots + \beta_p x_p$.
- Distribusi dari y mempunyai fungsi kepadatan dalam bentuk

$$f(y_i; \theta_i, \varphi) = \exp[A_i \{y_i \theta_i - \gamma(\theta_i)\} / \varphi + \tau(y_i, \varphi / A_i)]$$

dengan φ adalah suatu parameter skala (*scale parameter*), A_i adalah suatu bobot awal yang diketahui, dan parameter θ_i tergantung pada prediktor linear.

- Rata-rata atau mean, μ , adalah suatu fungsi invertibel yang halus dari prediktor linear, yaitu

$$\mu = m(\eta), \quad \text{dan} \quad \eta = m^{-1}(\mu) = l(\mu).$$

Fungsi invers, $l(\mu)$, disebut dengan suatu fungsi *link* atau *link function*.

Jika φ telah diketahui, maka distribusi dari y akan menjadi distribusi suatu keluarga eksponensial kanonik satu-parameter.

GLM membolehkan suatu perlakuan metodologi statistik untuk beberapa kelas penting dari model-model. Berikut ini adalah beberapa contoh penjabaran untuk beberapa distribusi yang termasuk dalam keluarga GLM.

○ **Gaussian atau Distribusi Normal**

Untuk Distribusi Normal maka $\varphi = \sigma^2$ dan dapat ditulis

$$\log f(y) = \frac{1}{\varphi} \left\{ y\mu - \frac{1}{2}\mu^2 - \frac{1}{2}y^2 \right\} - \frac{1}{2}\log(2\pi\varphi) ,$$

sehingga $\theta = \mu$ dan $\gamma(\theta) = \frac{\theta^2}{2}$.

○ **Distribusi Poisson**

Untuk Distribusi Poisson dengan mean μ diperoleh

$$\log f(y) = y \log \mu - \mu - \log(y!) ,$$

sehingga $\theta = \log \mu$, $\varphi = 1$ dan $\gamma(\theta) = \mu = e^\theta$.

○ **Distribusi Binomial**

Untuk Distribusi Binomial jumlah percobaan a dan parameter p , maka respon menjadi $y = s/a$ dengan s adalah jumlah sukses. Fungsi kepadatannya adalah

$$\log f(y) = a \left[y \log \frac{p}{1-p} + \log(1-p) \right] + \log \binom{a}{ay} ,$$

sehingga $A_i = a_i$, $\varphi = 1$, θ adalah transformasi logit dari p , dan

$$\gamma(\theta) = -\log(1-p) = \log(1+e^\theta) .$$

Keluarga GLM yang disediakan dalam **R-Commander** dengan fungsi **glm** mencakup distribusi **gaussian**, **binomial**, **poisson**, **Gamma**, **inverse.gaussian**, **quasibinomial**, dan **quasipoisson**.

Masing-masing distribusi dari respon memberikan suatu jenis fungsi *link* yang menghubungkan rata-rata atau mean dengan suatu prediktor linear. Fungsi-fungsi tersebut secara lengkap dapat dilihat pada Tabel 9.1 dengan **D** adalah notasi untuk **default** di **R**. Kombinasi dari distribusi respon dan fungsi *link* disebut dengan keluarga dari suatu GLM.

Untuk n pengamatan dari suatu GLM, fungsi log-likelihood yang dibentuk adalah

$$\log f(\theta, \varphi; Y) = \sum_i [A_i \{y_i \theta_i - \gamma(\theta_i)\} / \varphi + \tau(y_i, \varphi / A_i)]$$

dan mempunyai fungsi skor atau *score function* untuk θ yaitu

$$U(\theta) = A_i \{y_i - \gamma'(\theta_i)\} / \varphi .$$

Dari sini dapat ditunjukkan bahwa (secara lengkap lihat McCullagh dan Nelder (1989))

$$E(y_i) = \mu_i = \gamma'(\theta_i) \quad \text{dan} \quad \text{var}(y_i) = \frac{\phi}{A_i} \gamma''(\theta_i).$$

Tabel 9.1. Keluarga GLM dan fungsi link yang bersesuaian

Link	Nama Keluarga				
	binomial	Gamma	gaussian	inverse. gaussian	poisson
logit	D				
Probit	•				
cloglog	•				
identity		•	D		•
inverse		D			
log		•			D
1/ μ^2				D	
sqrt					•

Fungsi yang didefinisikan dengan $V(\mu) = \gamma''(\theta(\mu))$ disebut dengan suatu fungsi varians atau *variance function*. Untuk setiap distribusi respon, fungsi *link* $l = (\gamma')^{-1}$ untuk $\theta \equiv \eta$ disebut dengan suatu *link* kanonik atau *canonical link*. Tabel 9.2 berikut ini adalah daftar fungsi *canonical link* dan fungsi varians dari Keluarga GLM.

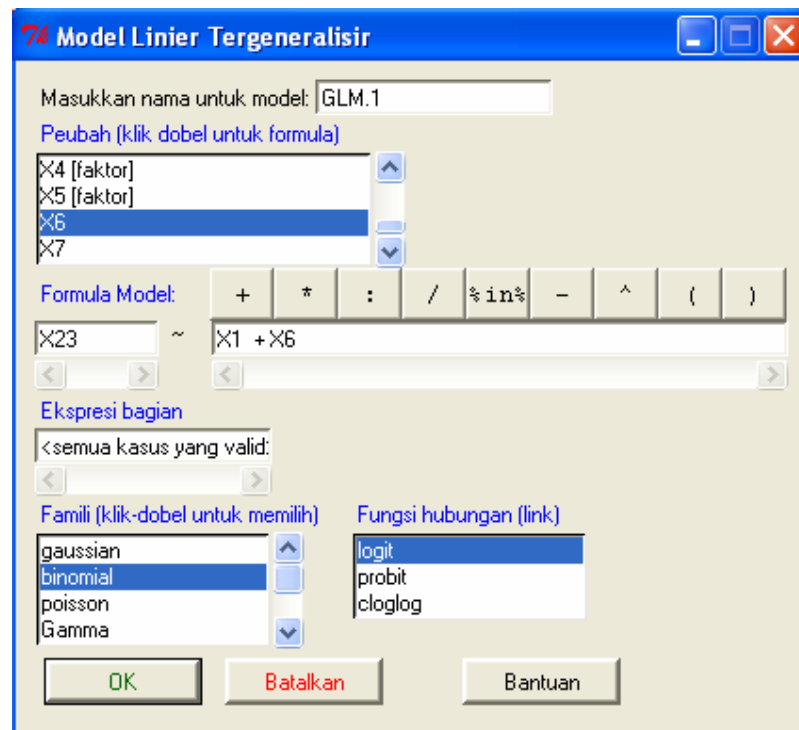
Tabel 9.2. Fungsi *canonical link* dan fungsi varians dari Keluarga GLM

Keluarga	Canonical link	Nama	Varians	Nama
binomial	$\log(\mu/(1-\mu))$	logit	$\mu/(1-\mu)$	$\mu(1-\mu)$
Gamma	$1/\mu$	invers	μ^2	μ^2
gaussian	μ	identitas	1	konstanta
Inverse.gaussian	$-2/\mu^2$	1/ μ^2	μ^3	μ^3
poisson	$\log \mu$	log	μ	μ

9.2. Contoh Kasus Model Linear Tergeneralisir dengan R-Commander

R-Commander menyediakan fasilitas untuk analisis **Model Linier Tergeneralisir** melalui menu **Statistika**, pilih **Pencocokan Model**, dan setelah itu pilih **Model Linier Tergeneralisir....** Secara umum, **Model Linier Tergeneralisir** digunakan untuk analisis model pada data variabel respon (dependen) yang mengikuti distribusi keluarga eksponensial. Ada beberapa keluarga distribusi yang tersedia di **R-Commander**, yaitu **gaussian**, **binomial**, **poisson**, **Gamma**, **inverse.gaussian**, **quasibinomial**, dan **quasi-poisson**. Pada bagian ini hanya dijelaskan pada kasus keluarga distribusi **binomial** yaitu variabel respon yang mempunyai dua kategori, sehingga model yang diperoleh dikenal dengan **model regresi logistik**.

Misalkan akan diteliti hubungan antara tipe konsumen berdasarkan lamanya menjadi konsumen (variabel **X1**) dan tingkat persepsi konsumen terhadap kualitas produk HBAT (variabel **X6**) terhadap kemauan konsumen untuk membangun hubungan dengan perusahaan di masa yang akan datang (variabel **X23**, yang jawabannya adalah **YA** dan **TIDAK**). Untuk keperluan analisis regresi logistik ini, pilih menu **Statistika**, pilih **Pencocokan Model**, dan kemudian pilih **Model Linier Tergeneralisir...**, sehingga diperoleh jendela dialog seperti berikut.



Gambar 9.1. Jendela dialog untuk analisis **Model Linier Tergeneralisir**

Untuk penyelesaian contoh kasus ini, ketik nama objek output model linear tergeneralisir (GLM) yang akan diestimasi (misal **GLM.1**). Hal ini berarti output hasil estimasi GLM disimpan sebagai objek dengan nama **GLM.1**. Kemudian pilih **X23** (kemauan untuk membangun hubungan di masa datang) pada sebelah kiri jendela **Formula Model:** (variabel dependen), dan pilih **X1 + X6** (jenis konsumen dan tingkat persepsi konsumen terhadap kualitas produk HBAT) pada jendela kanan dari **Formula Model** (variabel independen). Klik dua kali pada pilihan **binomial** di jendela **Famili**, dan pilih **Fungsi hubungan (link)** yang sesuai, yaitu **logit** pada kasus regresi logistik ini. Klik **OK** sehingga diperoleh output regresi logistik seperti berikut ini.

```
> GLM.1 <- glm(X23 ~ X1 + X6, family=binomial(logit), data=hbat)

> summary(GLM.1)

Call:
glm(formula = X23 ~ X1 + X6, family = binomial(logit), data = hbat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6566 -0.6071 -0.3165  0.8580  2.2036

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -4.6546    1.8036   -2.581  0.009858 **
X1[T.1 to 5 years]  2.9994    0.8111    3.698  0.000217 ***
X1[T.Over 5 years]  3.0377    0.9110    3.335  0.000854 ***
X6              0.2697    0.2245    1.201  0.229733
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 137.63  on 99  degrees of freedom
Residual deviance: 101.78  on 96  degrees of freedom
AIC: 109.78

Number of Fisher Scoring iterations: 5

> trellis.device(theme="col.whitebg")

> plot(all.effects(GLM.1), ask=FALSE)
```

Output di atas menunjukkan bahwa jenis (lama menjadi) konsumen HBAT berpengaruh signifikan terhadap kemauan konsumen untuk membangun hubungan di masa yang akan datang, sedangkan persepsi terhadap kualitas produk tidak berpengaruh terhadap kemauan untuk membangun hubungan di masa datang. Hal ini ditunjukkan oleh besarnya *p-value* dari uji **Z** pada kedua variabel *dummy* untuk **X1** tersebut yang lebih kecil dari $\alpha=0.05$. Sedangkan *p-value* dari uji **Z** untuk variabel **X6** lebih besar dari $\alpha=0.05$. Tanda koefisien regresi logistik yang positif pada kedua variabel *dummy* tersebut menjelaskan bahwa konsumen lama (**1-5 tahun** dan **lebih dari 5 tahun**) cenderung di masa datang mempunyai peluang yang lebih tinggi untuk membangun hubungan kembali dengan perusahaan dibanding konsumen baru (**kurang dari 1 tahun**).

Secara matematis, model regresi logistik yang diperoleh berdasarkan output di atas adalah

$$\hat{\mu}(x) = \frac{1}{1 + e^{-[-4,6546 + 2,9994X_{1_1} + 3,0377X_{1_2} + 0,2697X_6]}} ,$$

atau

$$\log\left(\frac{\hat{\mu}(x)}{1 - \hat{\mu}(x)}\right) = -4,6546 + 2,9994X_{1_1} + 3,0377X_{1_2} + 0,2697X_6 ,$$

dengan X_{1_1} dan X_{1_2} adalah variabel *dummy*, yaitu

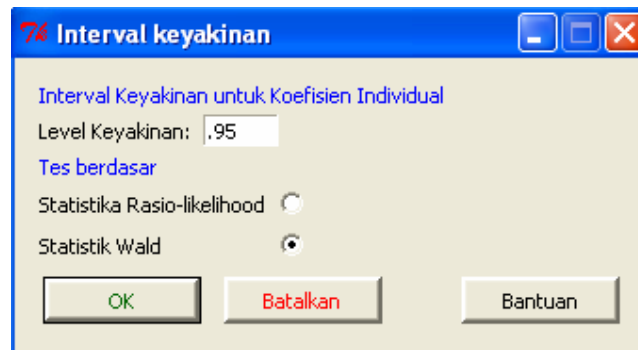
- X_{1_1} bernilai 1 untuk konsumen **1-5 tahun**, dan NOL untuk konsumen yang lain,
- X_{1_2} bernilai 1 untuk konsumen **lebih dari 5 tahun**, dan NOL untuk konsumen yang lain.

Odds Ratio adalah besaran yang biasanya digunakan dalam menginterpretasikan hasil suatu model regresi logistik. Secara lengkap bagaimana perhitungan *Odds Ratio* dan interpretasinya dapat dilihat di buku Hosmer dan Lemeshow (1989, hal. 40-47) yang berjudul **Applied Logistic Regression**.

Seperti pada bagian sebelumnya, **R-Commander** juga menyediakan fasilitas diagnostik numerik dan analisis grafik untuk evaluasi kebaikan model GLM yang telah diperoleh. Misalkan ingin ditampilkan interval keyakinan untuk koefisien model regresi logistik. Hal ini dapat dilakukan dengan cara memilih **Model**, dan kemudian klik **Interval keyakinan...**, sehingga muncul dialog pilihan seperti pada Gambar 9.2. Ada dua pilihan tes yang dapat ditampilkan interval keyakinannya sesuai dengan level keyakinan yang diinginkan, yaitu

- **Statistika Rasio-likelihood**, dan
- **Statistik Wald**.

Misalkan akan ditampilkan interval keyakinan dari **Statistik Wald**, maka klik pilihan pada **Statistik Wald**, dan kemudian klik **OK**.



Gambar 9.2. Jendela dialog **Interval keyakinan** pada GLM distribusi binomial

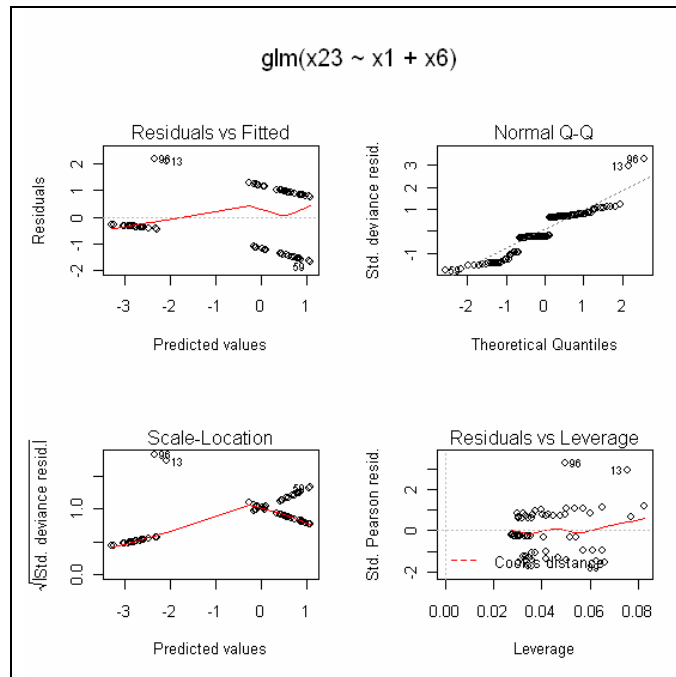
Output interval keyakinan dari **Statistik Wald** akan terlihat di jendela keluaran seperti berikut ini.

```
> Confint(GLM.1, level=.95, type="Wald")
```

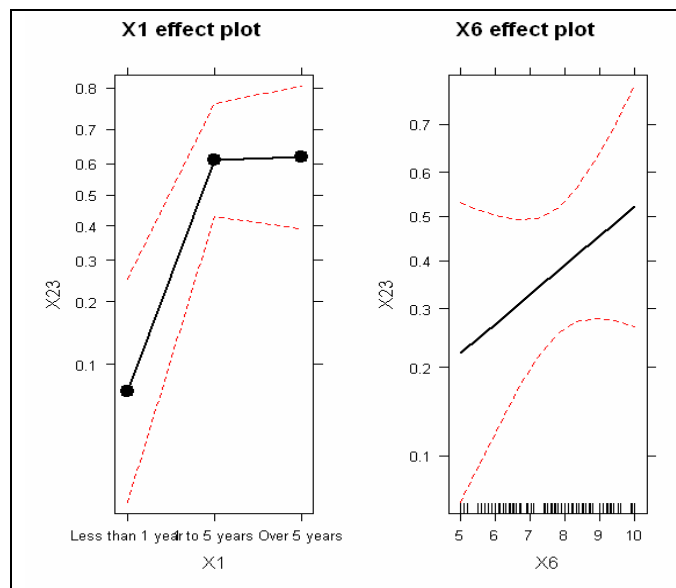
	2.5 %	97.5 %
(Intercept)	-8.189528	-1.1196538
X1[T.1 to 5 years]	1.409792	4.5890841
X1[T.Over 5 years]	1.252199	4.8231276
X6	-0.170395	0.7097233

Dari output interval keyakinan tersebut dapat dijelaskan bahwa variabel **X1** mempunyai pengaruh yang signifikan terhadap variabel respon, sedangkan variabel **X6** tidak berpengaruh terhadap variabel respon. Hal ini ditunjukkan dengan interval keyakinan koefisien dari **X1** yang tidak mencakup nilai NOL pada batas bawah dan atasnya.

Pada analisis grafik, ada beberapa pilihan grafik yang dapat ditampilkan untuk mengevaluasi kebaikan model GLM. Seperti pada model linear, beberapa grafik yang dapat ditampilkan adalah **Plot Diagnostik Dasar**, **Plot Komponen+Sisa**, **Plot Tambahan peubah**, **Plot Pengaruh**, dan **Plot Efek**. Pada **R-Commander**, plot-plot tersebut dapat dijalankan melalui menu **Model**, pilih **Grafik**, dan setelah itu pilih menu analisis grafik yang diinginkan. Jika pilihan grafik yang diinginkan adalah **Plot Diagnostik Dasar**, maka akan diperoleh output grafik yang terdiri dari empat macam plot seperti yang terlihat pada Gambar 9.3. Sebagai tambahan, jika pilihan grafik yang dipilih adalah **Plot Efek**, maka akan diperoleh output grafik yang terdiri dari dua macam plot seperti pada Gambar 9.4.



Gambar 9.3. Hasil Plot Diagnostik Dasar dari suatu GLM



Gambar 9.4. Plot Efek pada masing-masing variabel independen dari GLM

BAB 10

GRAFIK MENGGUNAKAN R-CLI

Secara garis besar ada dua cara untuk membuat grafik dalam **R**, yaitu dengan menggunakan **R-GUI** (lihat Bab 4) dan **R-CLI**. Pada bagian ini akan diberikan beberapa contoh pembuatan grafik dengan menggunakan **R-CLI**. Aktifkan jendela grafik terlebih dahulu sebelum membuat suatu grafik. Jika *user* memanggil suatu perintah pembuatan grafik, maka **R** secara otomatis akan mengaktifkan satu jendela grafik. Semua grafik yang dibuat akan di plot pada jendela grafik ini.

Jika *user* ingin mengaktifkan lebih dari satu jendela grafik dalam sistem operasi windows, *user* dapat mengaktifkan jendela grafik baru dengan perintah

```
> win.graph()
```

atau

```
> windows()
```

Untuk menutup jendela grafik terakhir yang sedang aktif, gunakan perintah

```
> graphics.off()
```

atau

```
> dev.off()
```

Secara umum perintah untuk pembuatan grafik didalam **R** dapat dikelompokkan menjadi 3 kelompok utama, yaitu

1. Fungsi-fungsi plot utama atau *high-level plotting commands*

Fungsi dalam kelompok ini dapat digunakan untuk membuat suatu plot baru pada jendela grafik. Beberapa fungsi tersebut adalah **plot**, **qqplot**, **hist**, **image**, **contour**, **persp**.

2. Fungsi-fungsi plot tambahan atau *low-level plotting commands*

Fungsi di kelompok ini dapat digunakan untuk menambahkan informasi tambahan kedalam suatu grafik yang telah dibuat dengan fungsi-fungsi plot utama diatas. Fungsi-fungsi tambahan ini dapat digunakan untuk menambahkan titik-titik baru, garis-garis atau keterangan-keterangan kedalam grafik. Beberapa fungsi yang termasuk kelompok ini adalah **points**, **lines**, **text**, **abline**, **legend**, **title**.

3. Fungsi-fungsi yang bersifat interaktif atau *interactive graphics functions*

Fungsi dalam kelompok ini memungkinkan *user* untuk menambahkan informasi atau mengambil informasi dari suatu plot yang telah ada menggunakan alat seperti **mouse**. Beberapa fungsi tersebut adalah **locator**, **identify**.

Paket **R** memiliki beberapa **library** yang berkaitan dengan pembuatan grafik. Pada bab ini pembahasan hanya difokuskan pada beberapa perintah yang berhubungan dengan pembuatan grafik pada **library** standar yaitu **graphics**. Ada banyak **library** lain yang dapat digunakan untuk pembuatan grafik, antara lain **aplpack**, **chplot**, **corrgram**, **gplot**, **grid**, **iplots**, **lattice**, **playwith**, **plotrix**, **rgl**, **Rgraphviz**, **Rgobi**, dan lain-lain.

Daftar dari perintah yang tersedia pada **library** standar **R** dapat dilihat dengan cara melihat nomor direktori dari library yang ada dalam sistem. Gunakan perintah **search()** untuk mengetahui nomor direktori tersebut (lihat hasil berikut ini).

```
> search()
[1] ".GlobalEnv"      "package:stats"  "package:graphics"
[4] "package:grDevices" "package:utils"  "package:datasets"
[7] "package:methods" "Autoloads"      "package:base"
```

Dari keluaran tersebut dapat dilihat bahwa library **graphics** sebagai **objects** berada di urutan ketiga dalam direktori search dari **R**. Kemudian untuk melihat daftar perintah dalam library **graphics** dapat digunakan perintah **objects** diikuti nomer urutan objek tersebut. Berikut adalah perintah-perintah yang ada dalam library **graphics** dengan menggunakan perintah **objects(3)**.

```
> objects(3)

[1] "abline"          "arrows"          "assocplot"       "axis"
[5] "Axis"            "axis.Date"       "axis.POSIXct"    "axTicks"
[9] "barplot"         "barplot.default" "box"             "boxplot"
[13] "boxplot.default" "bxp"             "cdplot"          "close.screen"
[17] "co.intervals"    "contour"         "contour.default" "coplot"
[21] "curve"           "dotchart"        "erase.screen"    "filled.contour"
[25] "fourfoldplot"    "frame"           "grid"            "hist"
[29] "hist.default"    "identify"        "image"           "image.default"
[33] "layout"          "layout.show"     "lcm"             "legend"
[37] "lines"           "lines.default"   "locator"         "matlines"
[41] "matplot"         "matpoints"       "mosaicplot"      "mtext"
[45] "pairs"           "pairs.default"   "panel.smooth"    "par"
[49] "persp"           "pie"             "piechart"        "plot"
[53] "plot.default"    "plot.design"     "plot.new"        "plot.window"
[57] "plot.xy"         "points"          "points.default"  "polygon"
[61] "rect"            "rug"             "screen"          "segments"
[65] "spineplot"       "split.screen"    "stars"           "stem"
[69] "strheight"       "stripchart"      "strwidth"        "sunflowerplot"
[73] "symbols"         "text"            "text.default"    "title"
[77] "xinch"           "xspline"         "xyinch"          "yinch"
```

Pada bagian berikut ini akan dibahas penggunaan perintah-perintah dalam library **graphics** diatas berdasarkan jenis kelompok perintah tersebut.

10.1. Fungsi-fungsi Plot Utama

Seperti yang dijelaskan sebelumnya, fungsi-fungsi plot utama dapat digunakan untuk membuat suatu plot baru dalam suatu jendela grafik. Jika jendela grafik yang sedang aktif telah berisi suatu grafik/plot, maka dengan perintah-perintah grafik tipe plot utama ini mengakibatkan **R** akan menghapus grafik/plot yang telah ada tersebut. Berikut ini adalah penjelasan beberapa fungsi yang termasuk dalam tipe plot utama ini.

10.1.1. Perintah `plot()`

Perintah **plot()** digunakan untuk menampilkan plot dari suatu data. Pada paket **R**, perintah plot ini dapat membuat plot/grafik yang bersesuaian dengan tipe dari data. Berikut ini adalah beberapa contoh penggunaan plot pada berbagai tipe data.

▪ Membuat diagram pencar atau *scatter plot* dari data *x* dan *y*

Perintah **plot(x,y)** dapat digunakan untuk membuat diagram pencar dari data *x* dan *y*. Perhatikan contoh data harga jual (*X* dalam ribu rupiah) dan volume penjualan atau *sales* (*Y* dalam juta rupiah) mingguan suatu produk pada tabel berikut ini.

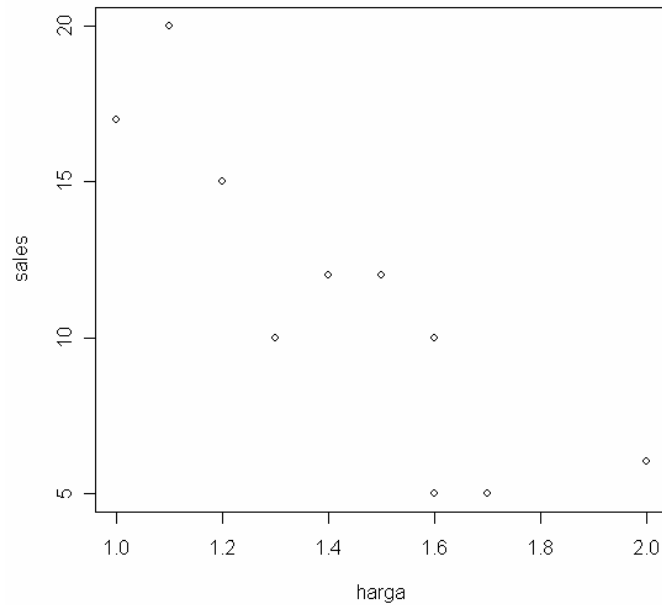
Tabel 10.1. Data harga dan *sales* selama 10 minggu pengamatan

Minggu ke	1	2	3	4	5	6	7	8	9	10
X (harga)	1.3	2	1.7	1.5	1.6	1.2	1.6	1.4	1	1.1
Y (<i>sales</i>)	10	6	5	12	10	15	5	12	17	20

Untuk membuat diagram pencar dari data harga dan *sales* tersebut, dapat digunakan script **R** berikut ini.

```
> harga=c(1.3,2,1.7,1.5,1.6,1.2,1.6,1.4,1,1.1)
> sales=c(10,6,5,12,10,15,5,12,17,20)
> plot(harga,sales)
```

Grafik keluaran dari perintah ini adalah diagram pencar seperti pada Gambar 10.1.



Gambar 10.1. Output diagram pencar dengan perintah **plot(x,y)**

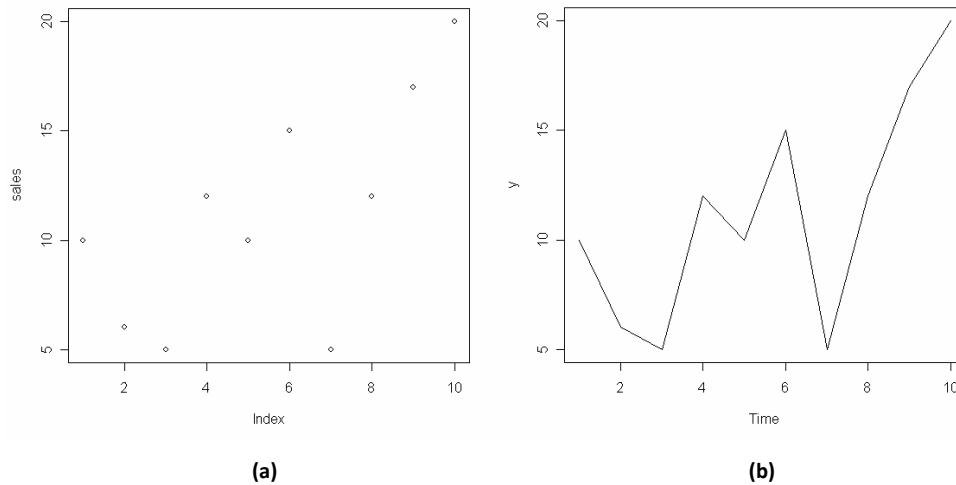
Berdasarkan diagram pencar diatas dapat dijelaskan bahwa ada hubungan linear negatif yang cukup kuat antara harga dan *sales*. Hal ini berarti jika terjadi kenaikan harga pada produk ada kecenderungan penjualan akan mengalami penurunan, dan sebaliknya.

▪ Membuat plot menurut indeks atau urutan waktu (Time Series Plot)

Perintah **plot(x)** dapat juga digunakan untuk membuat plot dari data x menurut indeks. Pada data yang bertipe runtun waktu (*time series*), perintah **plot(x)** akan menghasilkan plot dari x menurut urutan waktu atau dikenal dengan *Time Series Plot*. Perhatikan contoh pemakaian **plot(x)** pada data *sales* (dalam juta rupiah) mingguan suatu produk di Tabel 10.1 berikut ini.

```
> plot(sales)
> win.graph() # membuka jendela grafik baru untuk plot data
> y = ts(sales)
> plot(y)
```

Grafik keluaran dari perintah plot diatas dapat dilihat pada Gambar 10.2. Dari gambar ini dapat dijelaskan bahwa **plot(x)** menghasilkan *Time Series Plot* jika data bertipe runtun waktu (lihat gambar b).



Gambar 10.2. Output plot menurut indeks dengan perintah **plot(x)**

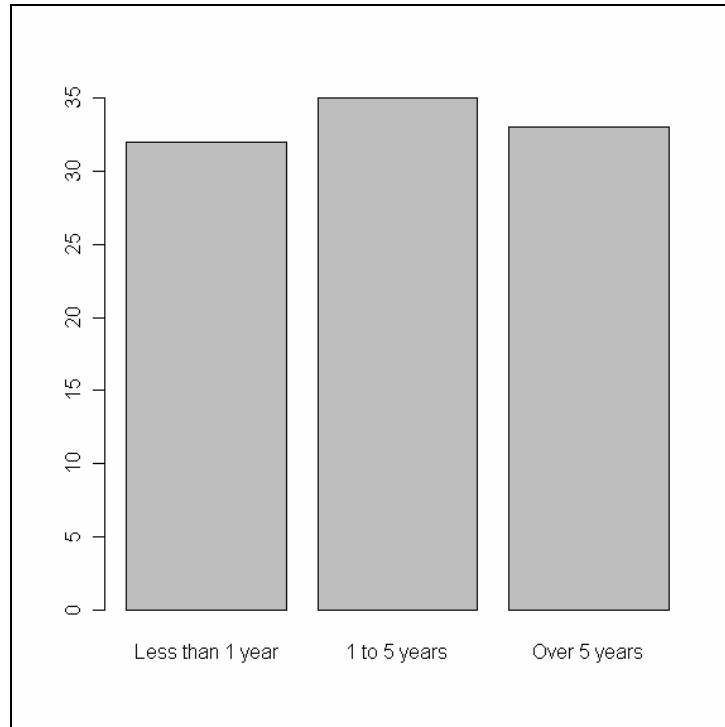
▪ Membuat plot dari data bertipe faktor

Pada Bab 7 sebelumnya diberikan contoh data HBAT yang mengandung data bertipe faktor. Untuk ilustrasi penggunaan perintah **plot(x)** pada data bertipe faktor, aktifkan kembali data HBAT yang sudah tersimpan di direktori **C:\Kerja_R** dalam file **R** dengan nama **hbat.RData**. Salah satu variabel yang bertipe faktor adalah **X1**, yaitu tipe konsumen berdasarkan lamanya menjadi konsumen HBAT. Perhatikan perintah-perintah berikut untuk memanggil data dan membuat plot pada variabel **X1**.

```
> load("C:\\Kerja_R\\hbat.RData")
> summary(hbat$X1)
  Less than 1 year   1 to 5 years   Over 5 years
                32             35             33

> plot(hbat$X1)
```

Grafik keluaran dari perintah plot untuk variabel **X1** diatas dapat dilihat pada Gambar 10.3 berikut ini.



Gambar 10.3. Output perintah **plot(x)** pada data bertipe faktor

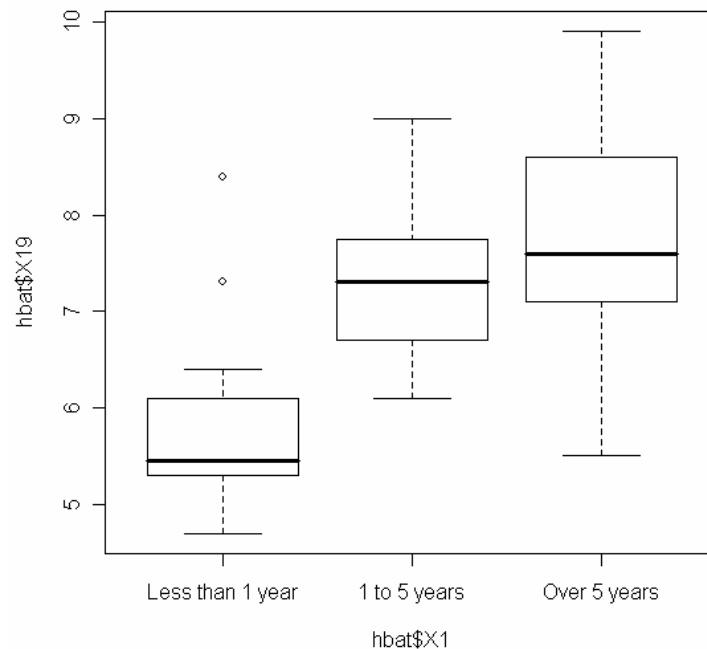
Berdasarkan grafik pada Gambar 10.3 dapat dijelaskan bahwa jumlah konsumen dengan lama menjadi konsumen 1-5 tahun adalah kelompok terbanyak dari 100 konsumen yang menjadi sampel, yaitu 35 konsumen.

Selain itu, perintah **plot()** juga dapat digunakan untuk membuat boxplot dari suatu variabel yang bersifat metrik berdasarkan suatu variabel nonmetrik (faktor). Misalkan akan dibuat boxplot tingkat kepuasan konsumen (**X19**) berdasarkan lama menjadi konsumen HBAT (**X1**). Perhatikan perintah dan hasil dari perintah berikut ini.

```
> numSummary(hbat[, "X19"], groups=hbat$X1, statistics=c("mean", "sd", "quantile"))
```

	mean	sd	0%	25%	50%	75%	100%	n
Less than 1 year	5.725000	0.7603055	4.7	5.35	5.45	6.10	8.4	32
1 to 5 years	7.314286	0.6983775	6.1	6.70	7.30	7.75	9.0	35
Over 5 years	7.654545	1.0779294	5.5	7.10	7.60	8.60	9.9	33

```
> plot(hbat$X19~hbat$X1)
```



Gambar 10.4. Output perintah **plot(x)** pada data metrik (**X19**) berdasarkan data yang bertipe faktor (**X1**)

10.1.2. Perintah **qqnorm(x)**, **qqline(x)**, **qqplot(x,y)**

Perintah-perintah ini digunakan untuk membuat dan menampilkan *Quantile-Quantile Plots* atau dikenal dengan **Q-Q plot**. Plot ini dapat digunakan untuk menguji apakah sekumpulan data berasal dari suatu distribusi tertentu, atau apakah dua sampel data memiliki distribusi yang identik (sama). Perintah **qqnorm** digunakan untuk menguji apakah suatu data mengikuti Distribusi Normal, sedangkan perintah **qqplot** dapat digunakan untuk membuat perbandingan dengan distribusi yang lain. Bersama dengan perintah **qqnorm**, perintah fungsi plot tambahan **qqline** dapat digunakan untuk menambahkan garis dari kuantil pertama ke kuantil ketiga dalam plot **qqnorm**. Data dapat dikatakan berasal dari distribusi yang bersifat *heavier tail* dibandingkan dengan Distribusi normal jika plot **qqnorm** memiliki bentuk turun (dibawah garis) pada bagian kiri dan naik (diatas garis) pada bagian kanan.

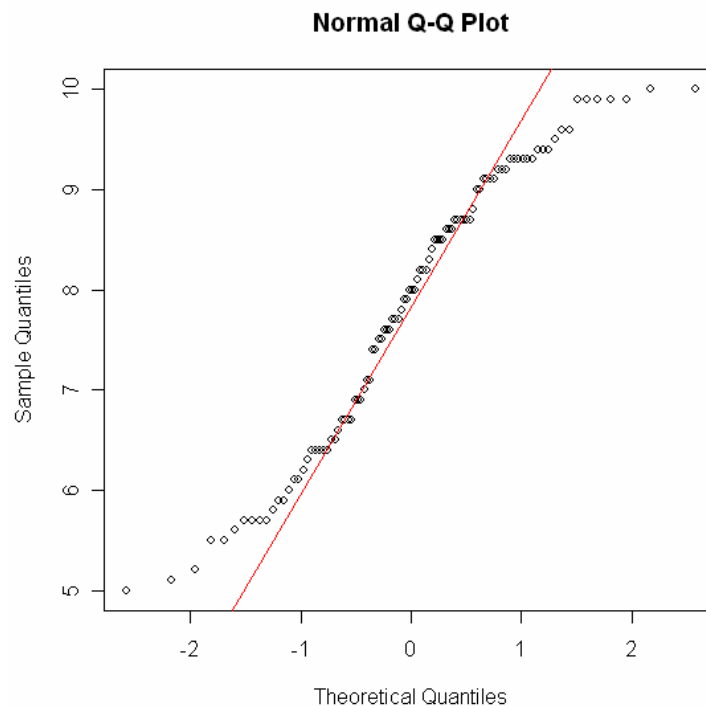
Berikut ini adalah contoh ilustrasi penggunaan **qqnorm** untuk uji kecocokan terhadap Distribusi Normal pada suatu variabel di data HBA, dan contoh penggunaan **qqplot** untuk perbandingan distribusi pada suatu data simulasi.

▪ Uji kecocokan terhadap Distribusi Normal

Untuk ilustrasi penggunaan perintah **qqnorm** pada suatu data, gunakan variabel **X6** (tingkat persepsi konsumen terhadap kualitas produk) pada data HBAT diatas. Perhatikan perintah-perintah berikut untuk membuat plot Kuantil-Kuantil Normal pada variabel **X6**.

```
> qqnorm(hbat$X6) # perintah untuk membuat Kuantil-Kuantil Normal  
> qqline(hbat$X6,col=2) # kuantil teoritis
```

Hasil dari perintah **qqnorm** dan **qqline** pada data **X6** diatas dapat dilihat pada Gambar 10.5. Dari gambar tersebut dapat dijelaskan bahwa secara visual data tidak berdistribusi Normal, karena terdapat sejumlah data dibagian kuantil atas dan bawah yang terletak diluar garis lurus.



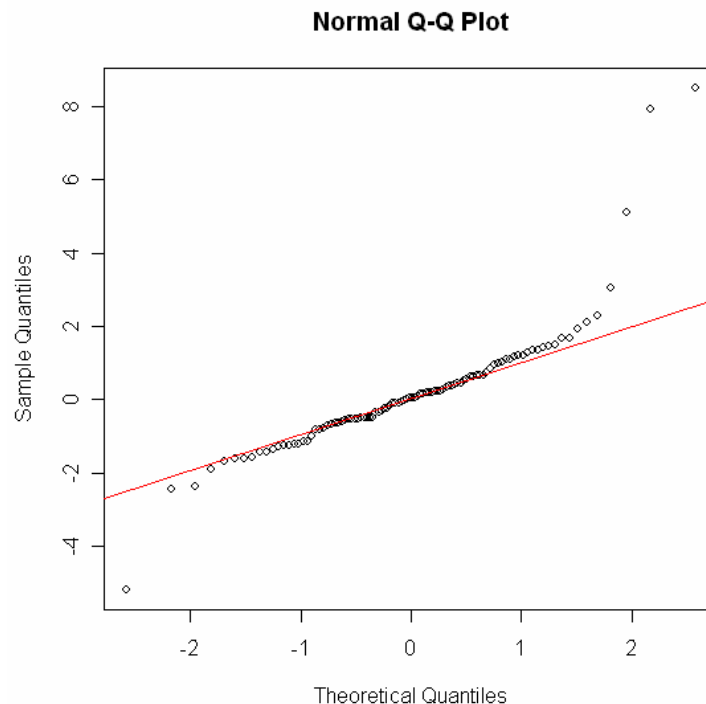
Gambar 10.5. Output perintah **qqnorm** dan **qqline** pada data **X6**

▪ Uji kecocokan terhadap Distribusi Statistik tertentu

Pada bagian ini akan diberikan ilustrasi penggunaan perintah **qqplot** untuk uji kecocokan terhadap distribusi tertentu pada suatu data sampel. Misalkan saja diketahui suatu data **y** yang dibangkitkan secara random (mengikuti distribusi **t** dengan **df=4**). Secara umum akan diperoleh data yang bersifat *heavy tail* karena dibangkitkan pada nilai **df** yang kecil. Perintah simulasi dan pengujian kenormalan data adalah sebagai berikut.

```
> y=rt(100,df=4)
> qqnorm(y)      # perintah untuk membuat Kuantil-Kuantil Normal
> qqline(y,col=2) # kuantil teoritis
```

Berikut adalah output dari perintah **qqnorm** dan **qqline** diatas.

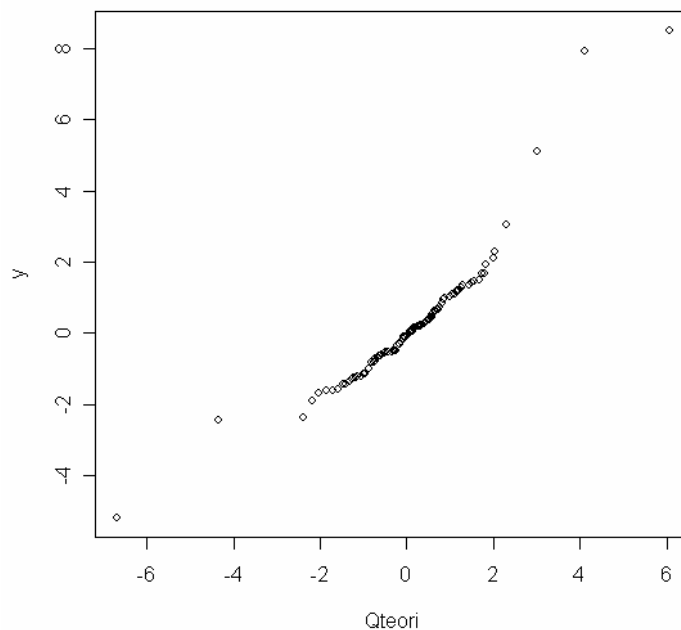


Gambar 10.6. Output perintah **qqnorm** dan **qqline** pada data **y**

Hasil dari perintah **qqnorm** dan **qqline** pada data **y** diatas menunjukkan bahwa data tidak berdistribusi Normal. Selanjutnya data akan dicoba bandingkan dengan distribusi **t** dengan **df** yang kecil. Untuk itu, bangkitkan sampel data lain dari distribusi **t** dengan **df 4**. Berikut perintah pembangkitan data dan qqplot untuk perbandingan distribusi.

```
> Qteori=rt(200,df=4)
> qqplot(Qteori,y) # distribusi teoritis pembeding sebagai x, data sebagai y
```

Output dari perintah diatas dapat dilihat pada Gambar 10.7. Dari gambar ini terlihat bahwa data secara visual relatif dalam garis lurus, sehingga distribusi **t** dengan **df=4** relatif cukup baik untuk memodelkan data simulasi **y** diatas.



Gambar 10.7. Output perintah **qqplot** pada data **y** dan **Qteori**

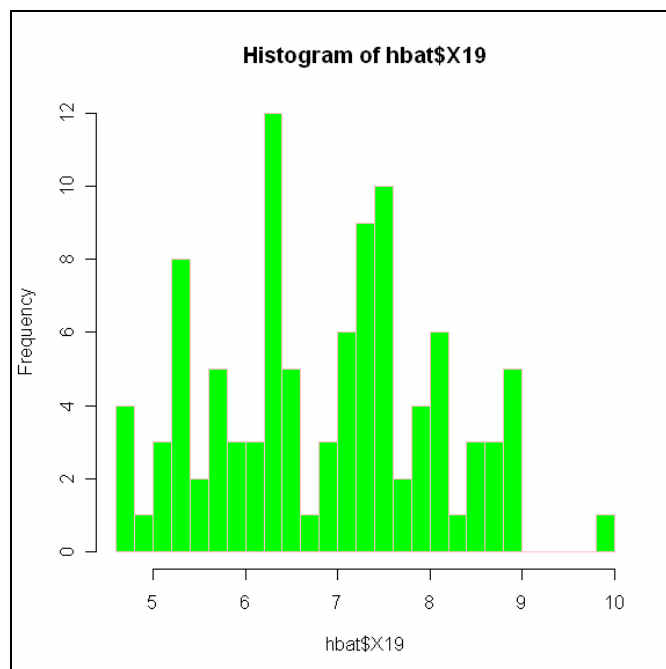
Sebagai catatan, karena sifat dari pengujian secara grafik yang cenderung subyektif, maka kesimpulan yang diperoleh harus dikonfirmasi dengan menggunakan uji statistik yang sesuai.

10.1.3. Perintah `hist(x)`

Perintah **hist** digunakan untuk membuat plot histogram dari suatu data tertentu. Perhitungan banyaknya kelas interval secara **default** di **R** menggunakan metode **Sturges**. Untuk pilihan lain yang tersedia berkaitan dengan pembuatan histogram dapat dilihat pada **help** perintah **hist**. Misalkan akan dibuat histogram dari variabel tingkat kepuasan konsumen HBAT atau variabel **X19**. Berikut ini adalah perintah-perintah untuk pembuatan histogram pada **X19**.

```
> hist(hbat$X19)
> hist(hbat$X19,breaks=20)
> hist(hbat$X19,breaks=20,col="green",border="pink")
> # Perhatikan perbedaan output histogram yang ditampilkan
```

Berikut ini adalah output histogram pada perintah **hist** yang terakhir, yaitu yang melibatkan argumen banyaknya kelas interval beserta warna histogramnya.



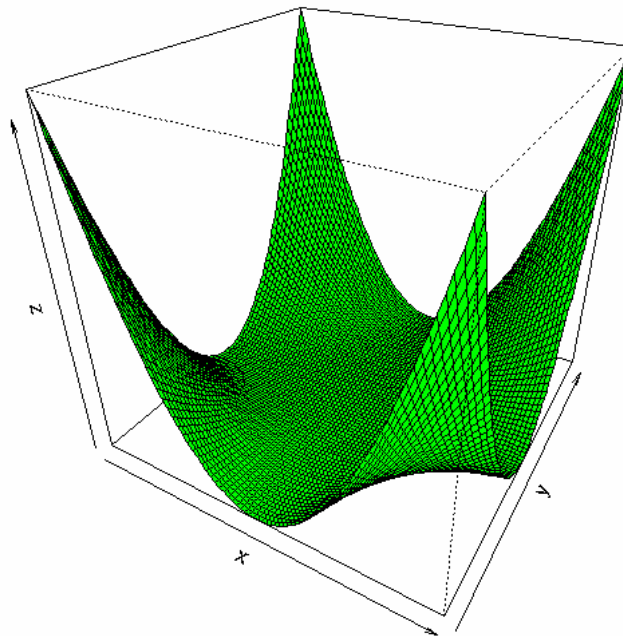
Gambar 10.8. Output perintah **hist** pada variabel **X19** data **hbat**

10.1.4. Perintah `image(x,y,z,...)`, `contour(x,y,z,...)`, `persp(x,y,z,...)`

Perintah **`persp`** adalah perintah yang digunakan untuk membuat plot tiga dimensi. Sedangkan perintah **`image`** dan **`contour`** digunakan untuk membuat plot proyeksi dua dimensi dari data tersebut. Untuk ilustrasi penggunaan ketiga perintah dan outputnya, perhatikan **script** berikut ini.

```
> y=seq(-20,20,0.5)    # bilangan antara -20 dan 20 dengan jarak 0.5
> x=seq(-20,20,0.5)
> z=outer(x^2,y^2,"*")  # z = x^2 + y^2 untuk semua elemen x dan y
> persp(x,y,z)          # plot dimensi tiga dari x, y, dan z dengan sudut default
> persp(x,y,z,theta=30,phi=30,col="green") # sudut dan warna beda
> # Perhatikan perbedaan output plot tiga dimensi yang ditampilkan
> image(x,y,z)
> contour(x,y,z)
```

Berikut ini adalah output dari perintah **`persp`** dan tambahan argumen-argumen diatas.



Gambar 10.9. Output perintah **`persp`** pada data simulasi

10.1.5. Argumen-argumen untuk fungsi plot utama

Secara lengkap argumen-argumen untuk fungsi plot utama dapat dilihat dengan perintah **help(plot)**. Berikut ini adalah beberapa argumen dan kegunaannya pada fungsi plot.

- **add=TRUE**

Argumen **add=TRUE** dapat digunakan untuk melakukan setting agar plot yang dibuat ditambahkan kedalam plot yang telah ada, yaitu fungsi plot utama. Sehingga plot yang dibuat bersifat seperti fungsi plot tambahan. Perintah ini hanya dapat digunakan untuk beberapa fungsi plot utama. Default nilai dari **add** adalah **add=FALSE**.

- **axes=FALSE**

Argumen **axes=FALSE** dapat digunakan untuk melakukan setting agar **axes** dari suatu plot tidak ditampilkan. Hal ini berguna apabila *user* akan membuat setting sendiri terhadap tampilan dari axis pada plot dengan perintah **axis()**. Default nilai dari **axes** adalah **axes=TRUE**, yaitu axis akan ditampilkan pada plot.

- **log="x", log="y", log="xy"**

Argumen ini bertujuan untuk merubah satuan dari sumbu x, y atau keduanya menjadi berskala **log**.

- **type=**

Argumen ini bertujuan untuk menentukan tipe dari plot yang dibuat. Berikut ini adalah beberapa pilihan tipe yang tersedia. (Catatan: contoh penggunaan adalah **type="l"** untuk membuat plot garis atau *lines*)

- "p" for points,
- "l" for lines,
- "b" for both,
- "c" for the lines part alone of "b",
- "o" for both 'overplotted',
- "h" for 'histogram' like (or 'high-density') vertical lines,
- "s" for stair steps,
- "S" for other steps, see *Details* below,
- "n" for no plotting.

- **xlab=string, ylab=string, main=string, sub=string**

Argumen ini bertujuan untuk memberi keterangan dari axis **x**, **y**, dan judul dari grafik. Argumen **sub** berfungsi untuk menampilkan subjudul, biasanya diletakkan dibawah axis **x**.

10.2. Fungsi-fungsi Plot Tambahan

Ada beberapa macam fungsi plot tambahan atau *low level graphics function* yang dapat digunakan untuk memperbaiki tampilan atau menambahkan sejumlah keterangan dalam plot yang telah dibuat dengan fungsi grafik utama. Berikut ini adalah penjelasan beberapa fungsi yang termasuk dalam tipe plot tambahan ini.

- **points(x,y)**

Fungsi atau perintah ini dapat digunakan untuk menambahkan titik-titik pada koordinat yang diberikan oleh **x** dan **y**.

- **lines(x,y)**

Perintah ini bertujuan untuk menambahkan garis menurut koordinat yang diberikan dalam **x** dan **y**.

- **text(x,y,labels,...)**

Perintah ini dapat digunakan untuk menambahkan suatu teks pada koordinat **x** dan **y**.

- **ablines(a,b), abline(h=y), abline(v=x)**

Perintah **ablines(a,b)** bertujuan untuk menambahkan garis lurus $y=a+bx$ pada plot yang telah ada. Sedangkan perintah **abline(h=y)** dan **abline(v=x)** digunakan untuk membuat garis horisontal atau vertikal sesuai dengan lokasi yang diberikan pada **h=y** atau **v=x**.

- **legend(x,y,legend,...)**

Perintah ini dapat digunakan untuk membuat **legend** dari suatu plot pada posisi yang diberikan koordinat **x** dan **y**. Beberapa argumen tambahan diberikan pada perintah **legend** (**v** menunjukkan suatu vektor yang bersesuaian nilainya dengan keterangan pada argumen **legend**), yaitu

- **legend(,lty=v)** untuk memberikan *line type* yang digunakan dalam plot
- **legend(,col=v)** untuk memberikan warna dari titik atau garis dalam plot
- **legend(,lwd=v)** untuk memberikan *line width* dari garis dalam plot

- **title(main,sub)**

Perintah ini bertujuan untuk memberikan judul dan subjudul dari plot. Hasil yang sama dapat diberikan dengan menggunakan argumen **main** dan **sub** dari fungsi plot utama.

10.3. Fungsi-fungsi Plot yang bersifat interaktif

Ada beberapa macam fungsi plot interaktif yang juga dapat digunakan untuk memperbaiki tampilan atau menambahkan sejumlah keterangan dalam plot yang telah dibuat dengan berinteraksi **R** menggunakan **mouse**. Berikut ini adalah penjelasan beberapa fungsi yang termasuk dalam tipe plot tambahan ini.

- **locator(n,type)**

Dengan perintah ini, **R** menunggu *user* untuk memilih **n** (maksimum 512) lokasi pada plot yang ada, dan membuat plot yang bersesuaian dengan spesifikasi yang diberikan pada argumen **type**.

- **locator()**

Perintah ini berguna untuk pemilihan lokasi dalam suatu plot secara interaktif, misalnya berguna untuk menempatkan **teks**, **label** atau **legend** pada posisi yang lebih tepat dalam grafik.

- **identify(x,y,labels)**

Perintah ini dapat digunakan untuk meletakkan label (atau nomer indeks dari data jika argumen label tidak diberikan) dari titik-titik yang diberikan dalam **x** dan **y**.

10.4. Notasi Matematika pada Plot

Sejumlah fasilitas untuk menambahkan simbol persamaan matematika kedalam suatu plot tersedia pada **R**. Informasi lengkap berkaitan dengan fasilitas ini dapat dilihat dengan perintah **help(plotmath)**. Untuk mengetahui beberapa contoh notasi hasil dari perintah **plotmath**, lakukan perintah-perintah berikut ini.

```
> help(plotmath)
> example(plotmath)  # Terdiri dari beberapa perintah dengan plotmath
> demo(plotmath)
```

Berikut ini adalah salah satu contoh pemakaian perintah **plotmath** untuk pembuatan persamaan matematika pada suatu plot data.

```

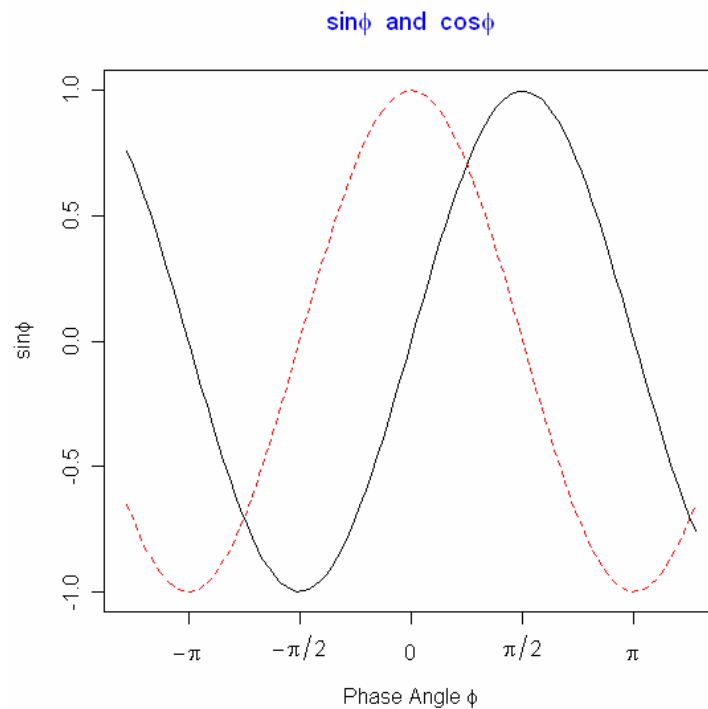
> x <- seq(-4, 4, len = 101)
> y <- cbind(sin(x), cos(x))

> matplot(x, y, type = "l", xaxt = "n",
+   main = expression(paste(plain(sin) * phi, " and ",
+     plain(cos) * phi)),
+   ylab = expression("sin" * phi, "cos" * phi), # only 1st is taken
+   xlab = expression(paste("Phase Angle ", phi)),
+   col.main = "blue")

> axis(1, at = c(-pi, -pi/2, 0, pi/2, pi),
+   labels = expression(-pi, -pi/2, 0, pi/2, pi))

```

Hasil dari **script** yang melibatkan perintah **matplot** diatas adalah sebagai berikut.



Gambar 10.10. Output perintah pembuatan notasi matematika pada plot

10.5. Setting parameter grafik

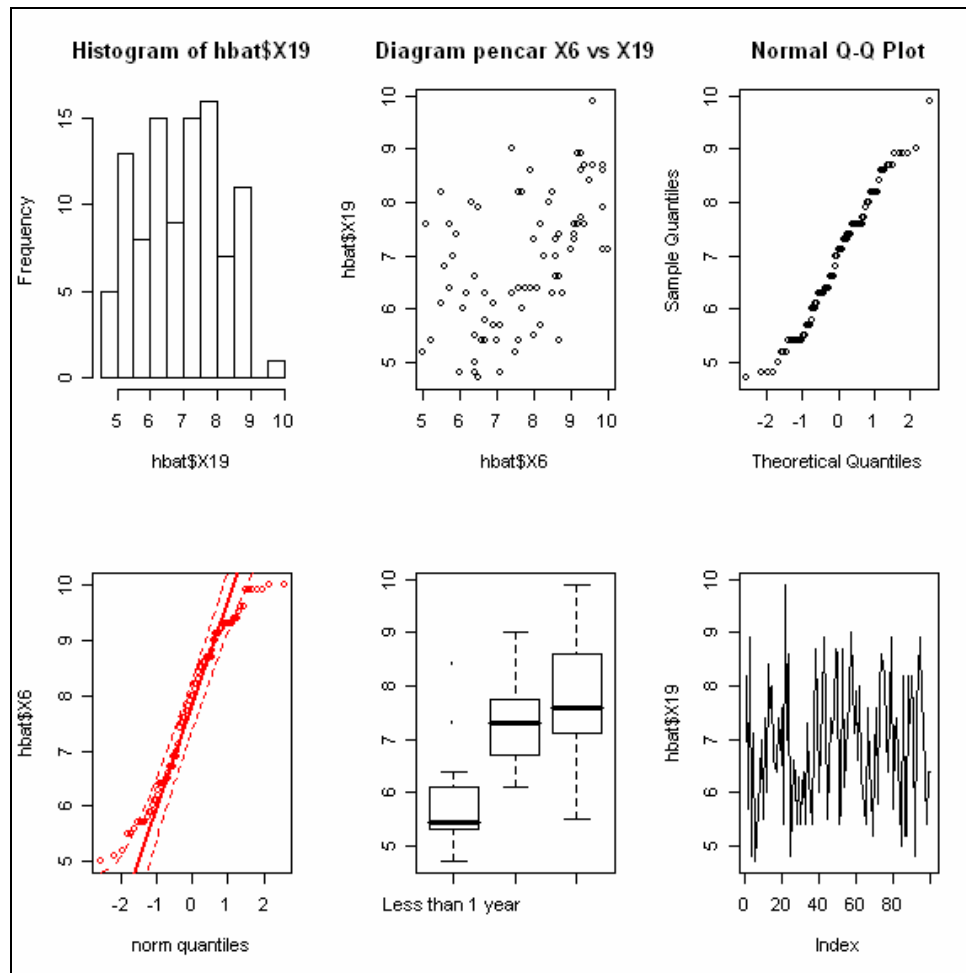
Default dari **R** dalam setiap jendela grafik hanya akan dibuat plot dari satu grafik. **Setting** ini dapat diubah sedemikian hingga dalam satu jendela grafik dapat dibuat lebih dari satu grafik, yaitu dengan menggunakan perintah **par**. Perintah **par** (singkatan dari kata **partisi**) ini diikuti dengan argumen **mfrow** (singkatan dari **multi figure row**). Misalkan akan dibuat 6 grafik dalam satu halaman, yaitu 2 baris dan 3 kolom, maka dapat digunakan perintah **par(mfrow=c(2,3))**. Berikut ini adalah contoh perintah untuk pembuatan 6 grafik dalam satu halaman dengan melibatkan variabel-variabel pada data **hbat.RData**.

```
> win.graph()
> par(mfrow=c(2,3))
> hist(hbat$X19) # Plot pertama sebuah histogram
> plot(hbat$X6,hbat$X19,main="Diagram pencar X6 vs X19") # Plot kedua
> qqnorm(hbat$X19) # Plot ketiga sebuah kuantil-kuantil normal dari X19
> qq.plot(hbat$X6) # Plot keempat sebuah kuantil-kuantil normal dari X6
> plot(hbat$X1,hbat$X19) # Plot kelima sebuah box-plot
> plot(hbat$X19,type="l") # Plot urutan indeks
```

Output dari perintah-perintah diatas dapat dilihat pada Gambar 10.11 di halaman 147. Hasil ini memberikan gambaran kepada *user* beberapa keunggulan pembuatan plot pada **R**, khususnya **multi plot** pada satu tampilan bersama-sama.

Keterangan lanjut berkaitan dengan setting dari parameter-parameter untuk grafik atau *Graphical Parameters* dapat dilihat pada menu help dari perintah **par**, yaitu dengan menggunakan perintah **?par**. Berikut adalah parameter-parameter tersebut.

- "ask",
- "fig", "fin",
- "lheight",
- "mai", "mar", "mex", "mfcol", "mfrow", "mfg",
- "new",
- "oma", "omd", "omi",
- "pin", "plt", "ps", "pty",
- "usr",
- "xlog", "ylog"



Gambar 10.11. Output perintah pembuatan plot dengan **setting parameter grafik**

BAB 11

ANALISIS RUNTUN WAKTU DENGAN R

Dalam dunia usaha yang terus menerus berubah dengan cepat, seorang manajer harus mampu menganalisis lingkungan yang terus berubah dan dapat memprediksi berbagai kemungkinan di masa depan. Kemampuan untuk meramal atau *forecast* masa depan usaha menjadi penting sebagai dasar pengambilan keputusan strategis bagi kelangsungan perusahaan. Sebagai contoh, bagian pemasaran suatu perusahaan yang ingin mengetahui permintaan suatu produk di masa mendatang, atau pemerintah ingin mengetahui dan memperkirakan berapa laju inflasi tahun-tahun mendatang.

Berbagai teknik untuk melakukan peramalan masa depan berdasarkan pada data masa lalu telah dikembangkan berdasarkan pada pengetahuan akan ilmu statistika. Secara umum ada dua pendekatan untuk peramalan, yaitu peramalan kuantitatif dan kualitatif. Peramalan kualitatif dilakukan jika data yang tersedia tidak ada atau tidak mencukupi, misalnya dalam proyek peluncuran produk baru. Metode peramalan kualitatif biasanya dilakukan secara subyektif, seperti teknik Delphi dan *expert opinion*.

Sedangkan metode peramalan kuantitatif dilakukan dengan menggunakan data masa lalu yang tersedia. Secara umum metode peramalan kuantitatif terbagi atas dua kelompok utama, yaitu : (Makridakis dkk., 1998; Hanke dan Reitsch, 2001)

- **Pendekatan Causal (sebab-akibat)**

Metode peramalan kelompok ini membahas proyeksi suatu kejadian berdasarkan variabel-variabel yang diduga mempengaruhi kejadian tersebut. Teknik peramalan yang termasuk pendekatan ini diantaranya adalah analisis regresi berganda, dan model ekonometrik.

- **Pendekatan Time Series**

Metode peramalan kelompok ini membahas proyeksi masa depan dari suatu variabel didasarkan pada data masa lalu dan sekarang.

Bab ini akan membahas penerapan paket **R** pada model peramalan dengan pendekatan time series yang banyak digunakan untuk melakukan kegiatan peramalan.

Data runtun waktu atau *time series* adalah data yang dikumpulkan, dicatat, atau diamati berdasarkan urutan waktu. Beberapa contoh time series adalah data bulanan tentang harga sembilan kebutuhan pokok, data bulanan mengenai konsumsi masyarakat akan daging ayam dan sapi, data bulanan tentang jumlah impor/ekspor komoditas tertentu, atau data harian dari Indeks Harga Saham Gabungan (IHSG) di Bursa Efek Jakarta, yang menunjukkan pergerakan IHSG setiap hari. Secara umum, tujuan dari analisis runtun waktu adalah untuk menemukan bentuk pola dari data di masa lalu dan menggunakan pengetahuan ini untuk melakukan peramalan terhadap sifat-sifat dari data di masa yang akan datang.

R menyediakan banyak library untuk analisis runtun waktu atau dikenal juga dengan *Time Series Analysis*. Selain pada library standar yaitu **stats**, analisis runtun waktu lebih lanjut dapat dilakukan dengan menggunakan library **fSeries**, **tseries**, **forecasting**, **strucchange**, **TSA**, **fArma**, **fracdiff**, dan masih banyak yang lain. Pada bagian ini akan dibahas penggunaan **R** untuk analisis model-model runtun waktu, seperti model tren, *exponential smoothing*, ARIMA, dan Neural Networks. Pembahasan tentang perintah-perintah di **R** untuk analisis runtun waktu di bab ini akan difokuskan pada penggunaan perintah di **R-Console**.

11.1. Model Trend Linear

Prinsip dari model Trend Linear adalah mencari persamaan trend linear dari data dan menggunakannya untuk mendapatkan ramalan pada waktu-waktu yang akan datang, \hat{Y}_{t+k} . Secara matematis, persamaan linear dari trend linear dapat ditulis sebagai berikut :

$$\hat{Y}_t = a + bt,$$

dengan a dan b adalah koefisien-koefisien persamaan linear yang akan dicari berdasarkan data yang ada, dan t adalah kode dari urutan periode waktu (biasanya $t=1,2,\dots$).

Sebagai contoh kasus, misalkan akan dilakukan peramalan jumlah penumpang pesawat udara internasional pada data **AirPassengers** yang sudah tersedia di **R**. Dengan menulis langsung nama data tersebut pada **R-Console**, maka akan diperoleh tampilan data runtun waktu mulai Januari 1949 sampai dengan Desember 1960 seperti berikut.

```
> AirPassengers
      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949 112 118 132 129 121 135 148 148 136 119 104 118
1950 115 126 141 135 125 149 170 170 158 133 114 140
1951 145 150 178 163 172 178 199 199 184 162 146 166
1952 171 180 193 181 183 218 230 242 209 191 172 194
1953 196 196 236 235 229 243 264 272 237 211 180 201
1954 204 188 235 227 234 264 302 293 259 229 203 229
1955 242 233 267 269 270 315 364 347 312 274 237 278
1956 284 277 317 313 318 374 413 405 355 306 271 306
1957 315 301 356 348 355 422 465 467 404 347 305 336
1958 340 318 362 348 363 435 491 505 404 359 310 337
1959 360 342 406 396 420 472 548 559 463 407 362 405
1960 417 391 419 461 472 535 622 606 508 461 390 432
```

Analisis tren linear dapat dilakukan dengan menggunakan perintah **lm** atau **linear model** seperti pada analisis regresi linear di Bab 7. Berikut adalah **script R** dan output persamaan tren linear pada data **AirPassengers**.

```
> AirPassengers
> t=1:length(AirPassengers)
> y=AirPassengers
> fit=lm(y~t)
> summary(fit)
```

Call:
lm(formula = y ~ t)

Residuals:

Min	1Q	Median	3Q	Max
-93.858	-30.727	-5.757	24.489	164.999

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	87.65278	7.71635	11.36	<2e-16 ***
t	2.65718	0.09233	28.78	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 46.06 on 142 degrees of freedom
Multiple R-squared: 0.8536, Adjusted R-squared: 0.8526
F-statistic: 828.2 on 1 and 142 DF, p-value: < 2.2e-16

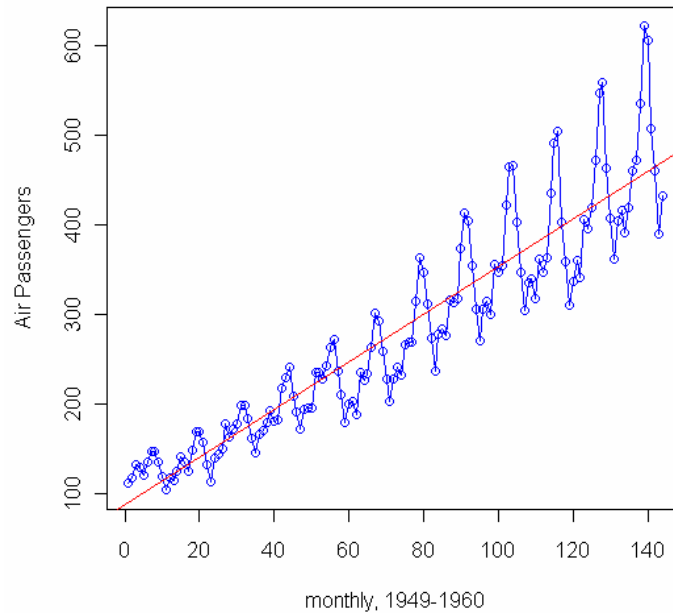
```
> plot(t,y,type="o", xlab="monthly, 1949-1960", ylab="Air Passengers")
> abline(fit)
```

Berdasarkan output perintah **summary(fit)** dapat dijelaskan bahwa persamaan tren linear untuk data **AirPassengers** adalah

$$\hat{Y}_t = 87,65278 + 2,65718 t .$$

Persamaan tren linear ini menunjukkan bahwa setiap bulan ada kenaikan jumlah penumpang pesawat udara internasional yaitu rata-rata sebesar 2,65718.

Dua perintah terakhir pada **script** diatas digunakan untuk membuat ilustrasi grafik yaitu plot antara data aktual dengan nilai-nilai prediksinya. Output dari perintah ini dapat dilihat pada Gambar 11.1 berikut ini.



Gambar 11.1. Output model tren linear pada data **AirPassengers**

11.2. Model Exponential Smoothing

Prinsip dari metode **Exponential Smoothing** adalah menggunakan nilai penghalusan secara eksponensial sebagai ramalan dari kejadian di satu waktu yang akan datang, \hat{Y}_{t+k} . Secara umum ada tiga macam model eksponensial, yaitu eksponensial **sederhana** (untuk data dengan pola stasioner), eksponensial **ganda** yang dikenal dengan model **Holt** (untuk data dengan pola tren), dan model **Holt-Winters** (untuk data dengan pola musiman dengan atau tanpa tren).

R menyediakan fasilitas untuk ketiga model tersebut dengan satu perintah yaitu **HoltWinters**. Penggunaan dari perintah ini adalah seperti berikut.

```
HoltWinters(x, alpha = NULL, beta = NULL, gamma = NULL,
seasonal = c("additive", "multiplicative"),
start.periods = 3, l.start = NULL, b.start = NULL,
s.start = NULL,
optim.start = c(alpha = 0.3, beta = 0.1, gamma = 0.1),
optim.control = list())
```

Perintah **HoltWinters** ini memiliki beberapa argumen yang dapat digunakan untuk menentukan pemilihan metode eksponensial smoothing mana yang akan dipilih. Berikut ini adalah argumen yang dapat dipilih pada perintah **HoltWinters**.

Argumen	Keterangan
<code>x</code>	An object of class <code>ts</code>
<code>alpha</code>	<i>alpha</i> parameter of Holt-Winters Filter.
<code>beta</code>	<i>beta</i> parameter of Holt-Winters Filter. If set to 0, the function will do exponential smoothing.
<code>gamma</code>	<i>gamma</i> parameter used for the seasonal component. If set to 0, a non-seasonal model is fitted.
<code>seasonal</code>	Character string to select an "additive" (the default) or "multiplicative" seasonal model. The first few characters are sufficient. (Only takes effect if <code>gamma</code> is non-zero).
<code>start.periods</code>	Start periods used in the autodetection of start values. Must be at least 3.
<code>l.start</code>	Start value for level (<code>a[0]</code>).
<code>b.start</code>	Start value for trend (<code>b[0]</code>).
<code>s.start</code>	Vector of start values for the seasonal component (<code>s_1[0]...s_p[0]</code>)
<code>optim.start</code>	Vector with named components <code>alpha</code> , <code>beta</code> , and <code>gamma</code> containing the starting values for the optimizer. Only the values needed must be specified. Ignored in the one-parameter case.
<code>optim.control</code>	Optional list with additional control parameters passed to <code>optim</code> if this is used. Ignored in the one-parameter case.

Model **Holt-Winters** yang disediakan di **R** terdiri dari dua pilihan, yaitu model aditif dan multiplikatif. Model aditif digunakan pada data runtun waktu dengan pola seasonal dengan variasi musiman konstan. Sedangkan model multiplikatif digunakan untuk data dengan pola seasonal yang mengandung variasi tidak konstan.

Fungsi prediksi pada model **Holt-Winters** aditif (untuk runtun waktu dengan panjang periode p) adalah

$$\hat{Y}_{t+h} = a[t] + h * b[t] + s[t + 1 + (h - 1) \bmod p],$$

dengan $a[t]$, $b[t]$ dan $s[t]$ adalah

$$a[t] = \alpha (Y[t] - s[t-p]) + (1-\alpha) (a[t-1] + b[t-1])$$

$$b[t] = \beta (a[t] - a[t-1]) + (1-\beta) b[t-1]$$

$$s[t] = \gamma (Y[t] - a[t]) + (1-\gamma) s[t-p].$$

Sedangkan fungsi prediksi pada model **Holt-Winters** multiplikatif (untuk runtun waktu dengan panjang periode p) adalah

$$\hat{Y}_{t+h} = (a[t] + h * b[t]) * s[t + 1 + (h - 1) \bmod p],$$

dengan $a[t]$, $b[t]$ dan $s[t]$ adalah

$$a[t] = \alpha (Y[t] / s[t-p]) + (1-\alpha) (a[t-1] + b[t-1])$$

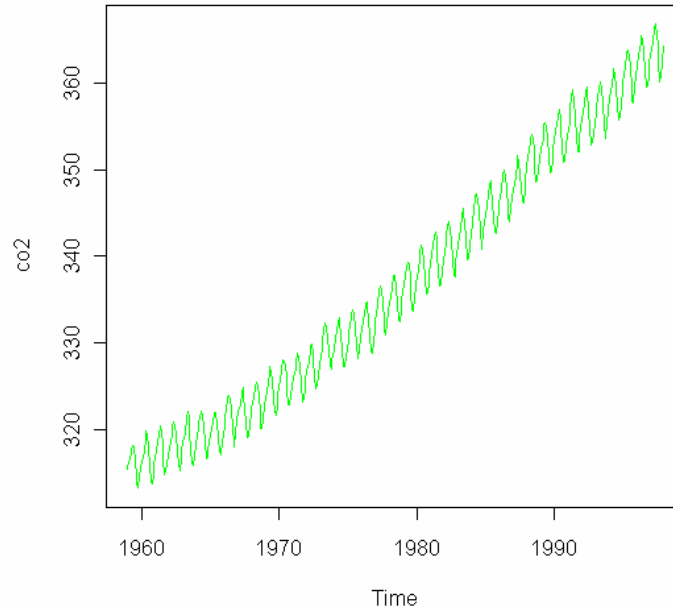
$$b[t] = \beta (a[t] - a[t-1]) + (1-\beta) b[t-1]$$

$$s[t] = \gamma (Y[t] / a[t]) + (1-\gamma) s[t-p].$$

Fungsi ini bekerja untuk mendapatkan nilai-nilai optimal dari α dan/atau β dan/atau γ dengan meminimalkan kuadrat dari error prediksi satu-tahap.

11.2.1. Model Holt-Winters Aditif

Sebagai contoh kasus, misalkan akan dilakukan peramalan CO_2 pada data **co2** yang sudah tersedia di **R**. Dengan menulis langsung nama data tersebut pada **R-Console**, maka akan diperoleh tampilan data runtun waktu mulai Januari 1959 sampai dengan Desember 1997. Sebagai tahap awal, identifikasi pola data dapat dilakukan dengan menampilkan plot **time series** dengan menggunakan perintah **plot(x)**. Output dari plot tersebut dapat dilihat pada Gambar 11.2. Dari gambar tersebut dapat dilihat bahwa data mengandung pola tren dan seasonal dengan variasi relatif konstan. Dengan demikian model Holt-Winters aditif adalah sesuai untuk diterapkan guna peramalan pada data.



Gambar 11.2. Output model tren linear pada data **AirPassengers**

Berikut adalah **script R** yang digunakan untuk menerapkan model Holt-Winters aditif pada data CO₂ yang sudah tersedia di **R**.

```
> # Seasonal Additive Holt-Winters
> co2
> plot(co2) # menampilkan plot time series dari data
> model1 <- HoltWinters(co2) # menerapkan Holt-Winters aditif (default)
> fore1 <- predict(model1, 50, prediction.interval = TRUE)
> plot(model1, fore1)
> plot(fitted(model1))
```

Script di atas menghasilkan koefisien-koefisien yang optimal pada model Holt-Winters aditif (**model1**), nilai ramalan 50 periode yang akan datang, dan grafik antara nilai aktual, prediksi dan ramalan 50 periode yang akan datang. Perintah terakhir menghasilkan plot tiap-tiap komponen data, yaitu **level**, **trend**, dan **seasonal**, serta **nilai ramalan**. Berikut ini adalah output lengkap pada masing-masing perintah diatas.

```

> model1
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = co2)

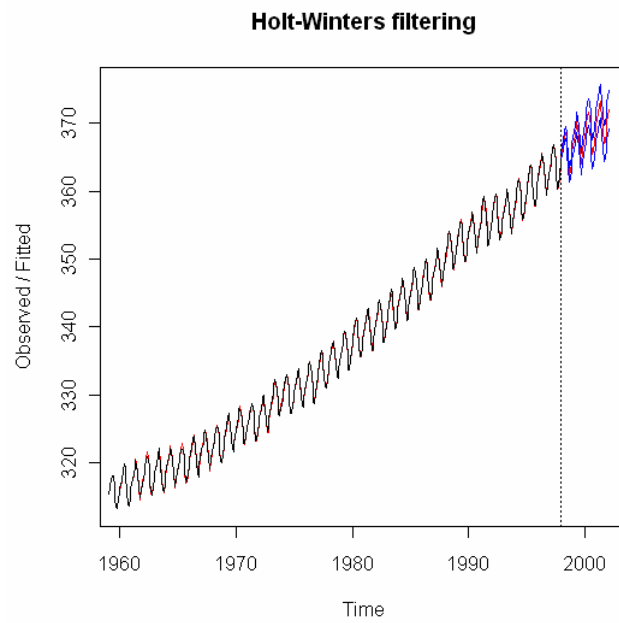
Smoothing parameters:
alpha: 0.4907075
beta : 0.01197529
gamma: 0.4536582

Coefficients:
      [,1]
a 364.6866567
b  0.1268701
s1 0.2812220
s2 1.0173743
s3 1.6642371
s4 2.9411121
s5 3.3487805
s6 2.5064789
s7 0.9613233
s8 -1.3122489
s9 -3.3464772
s10 -3.1988220
s11 -1.8558114
s12 -0.5254438

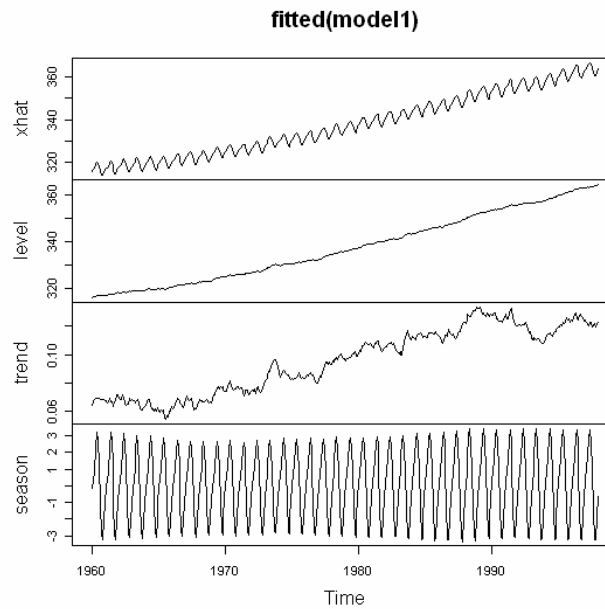
> fore1
      fit      upr      lwr
Jan 1998 365.0947 365.6900 364.4995
Feb 1998 365.9578 366.6224 365.2931
Mar 1998 366.7315 367.4603 366.0027
Apr 1998 368.1352 368.9244 367.3461
May 1998 368.6698 369.5162 367.8234
... ..
... ..
Oct 2001 367.3239 369.9810 364.6667
Nov 2001 368.7937 371.4892 366.0983
Dec 2001 370.2510 372.9848 367.5171
Jan 2002 371.1845 373.9829 368.3861
Feb 2002 372.0475 374.8841 369.2110

```

Output diatas menunjukkan bahwa nilai parameter smoothing yang optimal adalah **alpha=0.49**, **beta=0.01**, dan **gamma=0.45**. Koefisien-koefisien **a** dan **b**, serta koefisien **seasonal (s1,s2,...,s12)** juga diberikan. Selanjutnya juga diberikan nilai-nilai ramalan 50 periode kedepan beserta taksiran batas atas dan bawah. Pada akhirnya plot komponen data, serta perbandingan antara nilai aktual dan ramalan ditampilkan seperti pada Gambar 11.3 dan 11.4



Gambar 11.3. Nilai aktual dan ramalan pada data CO_2



Gambar 11.4. Nilai-nilai komponen **level**, **trend**, dan **seasonal** pada data CO_2

11.2.2. Model Holt-Winters Multiplikatif

Misalkan akan dilakukan peramalan jumlah penumpang pesawat udara pada data **AirPassengers** yang sudah tersedia di **R**. Sebagai tahap awal, identifikasi pola data dapat dilakukan dengan menampilkan plot **time series** (lihat Gambar 11.1) menunjukkan bahwa data mengandung pola tren dan seasonal dengan variasi yang cenderung meningkat. Dengan demikian salah satu model yang sesuai untuk peramalan pada data **AirPassengers** ini adalah model Holt-Winters multiplikatif.

Berikut adalah **script R** yang digunakan untuk menerapkan model Holt-Winters multiplikatif pada data **AirPassengers** yang sudah tersedia di **R**.

```
> # Seasonal Multiplicative Holt-Winters
> AirPassengers
> plot(AirPassengers) # menampilkan plot time series dari data
> model2 <- HoltWinters(AirPassengers, seasonal="mult")
> fore2 <- predict(model2, 24, prediction.interval = TRUE)
> plot(model2,fore2)
> plot(fitted(model2))
```

Script di atas menghasilkan koefisien-koefisien yang optimal pada model Holt-Winters multiplikatif (**model2**), nilai ramalan 24 periode yang akan datang, dan grafik antara nilai aktual dan ramalan 24 periode yang akan datang. Perintah terakhir menghasilkan plot tiap-tiap komponen data, yaitu **level**, **trend**, dan **seasonal**, serta **nilai ramalan**. Berikut ini adalah sebagian output pada perintah-perintah diatas.

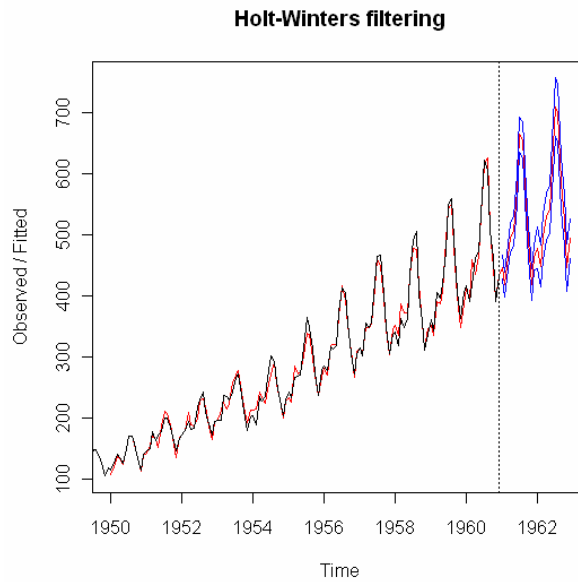
```
> model2
Holt-Winters exponential smoothing with trend and multiplicative seasonal component.

Call:
HoltWinters(x = AirPassengers, seasonal = "mult")

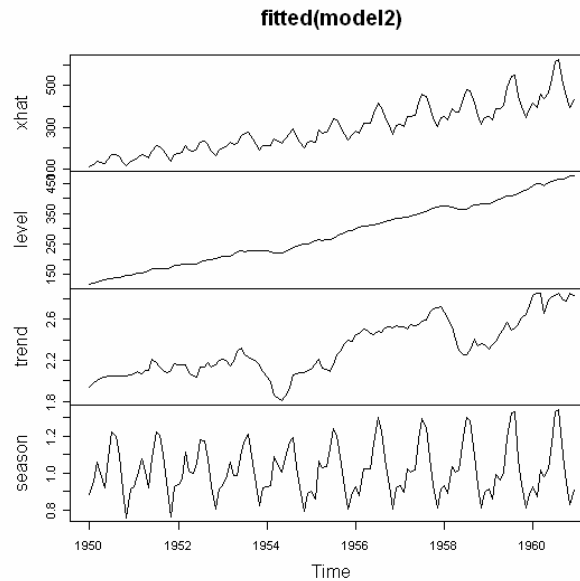
Smoothing parameters:
alpha: 0.274855
beta : 0.01745283
gamma: 0.8766261

Coefficients:
      [,1]
a 475.6200718 ... dan seterusnya.
```

Plot komponen data, serta perbandingan antara nilai aktual dan ramalan ditampilkan seperti pada Gambar 11.5 dan 11.6 berikut ini.



Gambar 11.5. Nilai aktual dan ramalan pada data **AirPassengers**



Gambar 11.6. Nilai komponen **level**, **trend**, dan **seasonal** pada **AirPassengers**

11.2.3. Model Holt-Winters Non-seasonal atau Model Eksponensial Ganda

Misalkan akan dilakukan peramalan jumlah populasi penduduk United States (dalam juta jiwa) pada data **uspop** yang sudah tersedia di **R** (ditambah suatu error). Tahap awal identifikasi pola data menunjukkan bahwa data mengandung pola tren yang cenderung meningkat. Dengan demikian salah satu model yang sesuai untuk peramalan pada data **uspop** ini adalah model eksponensial ganda atau Holt-Winters non-seasonal.

Berikut adalah **script R** yang digunakan untuk menerapkan model Holt-Winters non-seasonal pada data **uspop** yang sudah tersedia di **R**, serta beberapa output dari perintah-perintah tersebut.

```
> # Non-Seasonal Holt-Winters
> uspop
Time Series:
Start = 1790
End = 1970
Frequency = 0.1
[1] 3.93 5.31 7.24 9.64 12.90 17.10 23.20 31.40 39.80 50.20
[11] 62.90 76.00 92.00 105.70 122.80 131.70 151.30 179.30 203.20

> x <- uspop + rnorm(uspop, sd = 5) # error N(0,5) ditambahkan ke data uspop
> model3 <- HoltWinters(x, gamma = 0) # gamma=0 untuk eksponensial ganda
> model3
Holt-Winters exponential smoothing with trend and without seasonal component.

Call:
HoltWinters(x = x, gamma = 0)

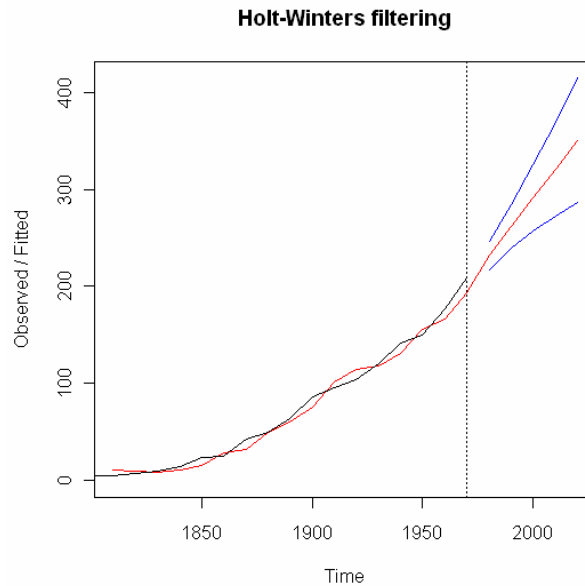
Smoothing parameters:
alpha: 0.5747944
beta : 1
gamma: 0

Coefficients:
[1]
a 201.89779
b 29.71546

> model3$SSE # nilai SSE dari model Holt-Winters non-seasonal
[1] 1020.530

> fore3 <- predict(model3, 5, prediction.interval = TRUE) # ramalan 5 tahun kedepan
> plot(model3,fore3)
```

Plot perbandingan antara nilai aktual dan ramalan dari model Holt-Winter non-seasonal pada data **uspop** adalah sebagai berikut.



Gambar 11.7. Nilai aktual dan ramalan pada data **uspop**

11.2.4. Model Eksponensial Smoothing Sederhana

Misalkan saja akan dilakukan penerapan model eksponensial smoothing sederhana untuk peramalan jumlah populasi penduduk United States (dalam juta jiwa) pada data **uspop** yang sudah tersedia di **R**, seperti pada bagian sebelumnya. Berikut adalah **script R** yang digunakan untuk menerapkan model eksponensial smoothing sederhana pada data **uspop** (plus suatu error), serta beberapa output dari perintah-perintah tersebut.

```
> # Exponential Smoothing
> uspop
> x <- uspop + rnorm(uspop, sd = 5)
> model4 <- HoltWinters(x, gamma = 0, beta = 0)
> fore4 <- predict(model4, 5, prediction.interval = TRUE) # ramalan 5 tahun kedepan
```

```

> model4
Holt-Winters exponential smoothing without trend and without seasonal component.

Call:
HoltWinters(x = x, beta = 0, gamma = 0)

Smoothing parameters:
alpha: 0.9999216
beta : 0
gamma: 0

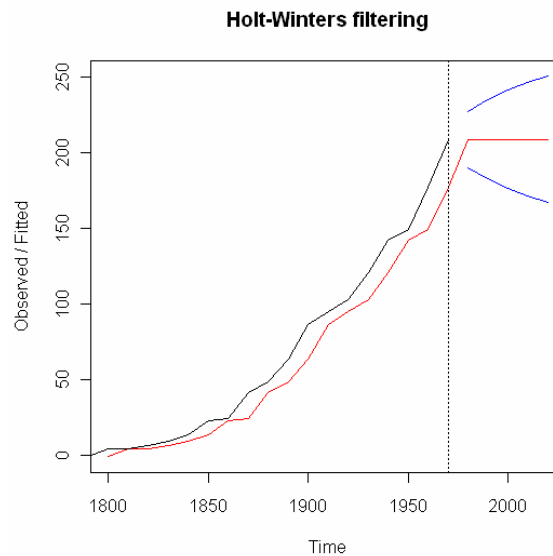
Coefficients:
[,1]
a 208.6348

> model4$SSE # menampilkan nilai SSE dari model eksponensial smoothing sederhana
[1] 3974.916

> plot(model4,fore4)

```

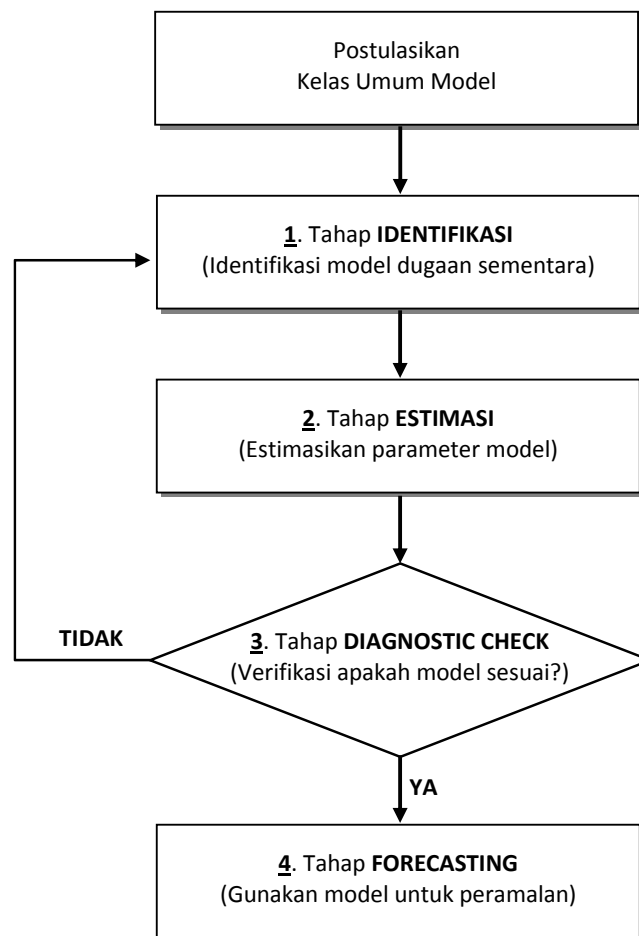
Plot perbandingan antara nilai aktual dan ramalan dari model eksponensial smoothing sederhana pada data **uspop** adalah sebagai berikut.



Gambar 11.8. Nilai aktual dan ramalan pada data **uspop** dengan metode eskponensial smoothing sederhana

11.3. Model ARIMA

Model *Autoregressive Integrated Moving Average* (ARIMA) merupakan salah satu model yang populer dalam peramalan dengan pendekatan time series. Model ini terdiri dari tiga bentuk utama yaitu model AR, MA, dan ARMA. Prosedur Box-Jenkins adalah suatu prosedur standar yang banyak digunakan dalam pembentukan model ARIMA. Prosedur ini terdiri dari empat tahapan yang iteratif dalam pembentukan model ARIMA pada suatu data runtun waktu, yaitu tahap identifikasi, estimasi, *diagnostic check*, dan peramalan. Berikut ini adalah diagram yang menggambarkan tahap-tahap dalam prosedur Box-Jenkins (Bowerman dan O'Connell, 1993; Wei, 2006).



Gambar 11.9. Prosedur Box-Jenkins untuk pembentukan model ARIMA

Secara umum, bentuk matematis dari model ARIMA(p,d,q) dapat ditulis sebagai berikut (Cryer, 1986; Wei, 2006)

$$(1 - \phi_1 B - \dots - \phi_p B^p)(1 - B)^d Z_t = \theta_0 + (1 - \theta_1 B - \dots - \theta_q B^q) a_t,$$

dengan B adalah operator mundur, yaitu $B^k Z_t = Z_{t-k}$. Penentuan orde p dan q dari model ARIMA pada suatu data runtun waktu dilakukan dengan mengidentifikasi plot *Autocorrelation Function* (ACF) dan *Partial Autocorrelation Function* (PACF) dari data yang sudah stasioner. Berikut ini adalah petunjuk umum untuk penentuan orde p dan q pada suatu data runtun waktu yang sudah stasioner.

Tabel 11.1. Pola teoritis ACF dan PACF dari proses yang stasioner

Proses	ACF	PACF
AR(p)	<i>Dies down</i> (turun cepat secara eksponensial / sinusoidal)	<i>Cuts off after lag p</i> (terputus setelah lag p)
MA(q)	<i>Cuts off after lag q</i> (terputus setelah lag q)	<i>Dies down</i> (turun cepat secara eksponensial / sinusoidal)
ARMA(p,q)	<i>Dies down</i> (turun cepat secara eksponensial / sinusoidal))	<i>Dies down</i> (turun cepat secara eksponensial / sinusoidal))
AR(p) atau MA(q)	<i>Cuts off after lag q</i> (terputus setelah lag q)	<i>Cuts off after lag p</i> (terputus setelah lag p)
<i>White noise</i> (Random)	Tidak ada yang signifikan (tidak ada yang keluar batas)	Tidak ada yang signifikan (tidak ada yang keluar batas)

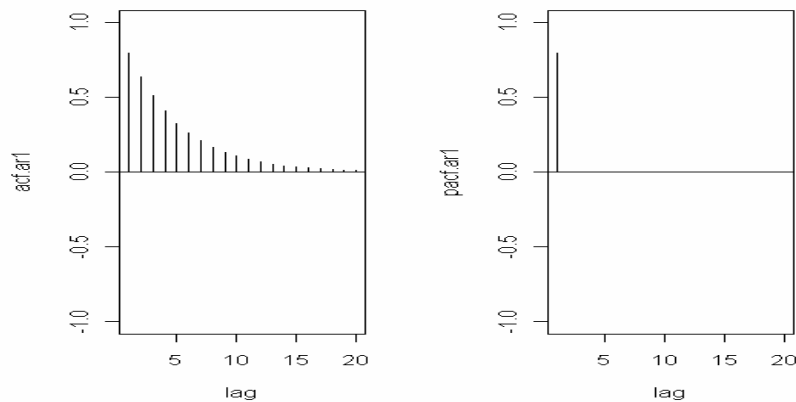
R adalah salah satu paket statistika yang menyediakan fasilitas untuk membuat bentuk ACF dan PACF teoritis dari model-model ARIMA yang stasioner. Berikut ini adalah contoh **script** untuk membuat plot ACF dan PACF teoritis dari model AR(p), MA(q) dan ARMA(p,q), serta outputnya.

```

> # ACF dan PACF teoritis untuk AR(1)
> acf.ar1 = ARMAacf(ar=0.8, ma=0, 20)
> pacf.ar1 = ARMAacf(ar=0.8, ma=0, 20, pacf=T)
> acf.ar1 = acf.ar1[2:21]
> c1 = acf.ar1
> c2 = pacf.ar1
> ar1 = cbind(c1, c2)
> ar1 # Nilai-nilai ACF dan PACF teoritis
> par(mfrow=c(1,2))
> plot(acf.ar1, type="h", xlab="lag", ylim=c(-1,1))
> abline(h=0)
> plot(pacf.ar1, type="h", xlab="lag", ylim=c(-1,1))
> abline(h=0)

```

Berikut ini adalah hasil plot ACF dan PACF teoritis dari model ARIMA(1,0,0) yang biasanya disingkat model AR(1), dengan nilai koefisien parameter model (ϕ) 0,8.



Gambar 11.10. Plot ACF dan PACF teoritis model AR(1) dengan $\phi = 0,8$

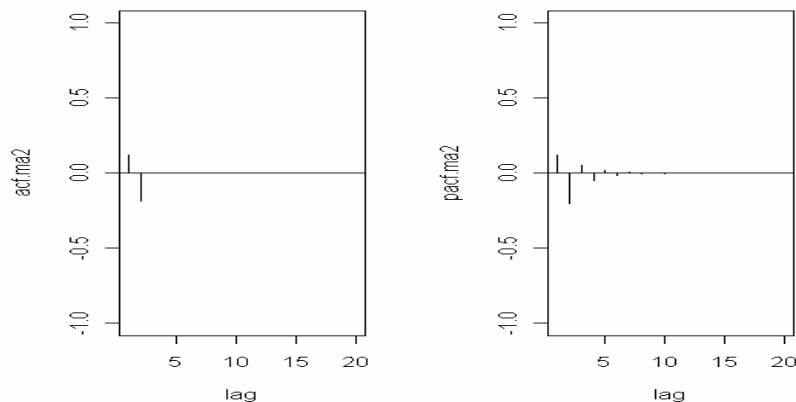
Dari gambar diatas dapat dijelaskan bahwa plot ACF pada model AR(1) dengan koefisien parameter positif adalah *dies down* (turun cepat secara eksponensial) dengan nilai ACF yang selalu positif. Sedangkan PACF menunjukkan pola yang terputus setelah lag 1 seperti petunjuk pada Tabel 11.1.


```

> # ACF dan PACF teoritis untuk MA(2)
> acf.ma2 = ARMAacf(ar=0, ma=c(1.5, -0.7), 20)
> pacf.ma2 = ARMAacf(ar=0, ma=c(1.5, -0.7), 20, pacf=T)
> acf.ma2 = acf.ma2[2:21]
> c1 = acf.ma2
> c2 = pacf.ma2
> ma2 = cbind(c1, c2)
> ma2 # Nilai-nilai ACF dan PACF teoritis
> par(mfrow=c(1,2))
> plot(acf.ma2, type="h", xlab="lag", ylim=c(-1,1))
> abline(h=0)
> plot(pacf.ma2, type="h", xlab="lag", ylim=c(-1,1))
> abline(h=0)

```

Di bawah ini adalah hasil plot ACF dan PACF teoritis dari model ARIMA(0,0,2) yang biasanya disingkat model MA(2), dengan nilai koefisien parameter model (*tetha* 1 dan 2) 1,5 dan -0,7.



Gambar 11.10. Plot ACF dan PACF teoritis model MA(2) dengan *tetha* 1,5 dan -0,7

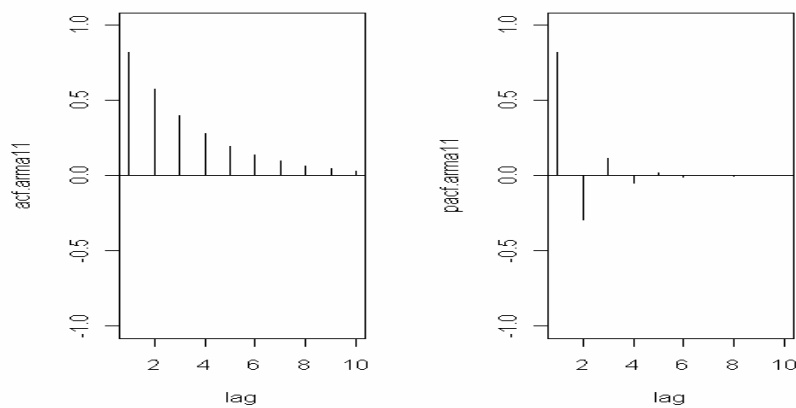
Berdasarkan pola pada gambar diatas dapat dijelaskan bahwa plot ACF pada model MA(2) dengan koefisien parameter positif 1,5 (*tetha1*) dan -0,7 (*tetha2*) adalah terputus setelah lag 2. Sedangkan PACF menunjukkan pola yang *dies down* (turun cepat secara sinusoidal) dengan nilai PACF yang berubah dari positif ke negatif seperti petunjuk pada Tabel 11.1 diatas.

```

> # ACF dan PACF teoritis untuk ARMA(1,1)
> acf.arma11 = ARMAacf(ar=0.7, ma=0.4, 10)
> pacf.arma11 = ARMAacf(ar=0.7, ma=0.4, 10, pacf=T)
> acf.arma11 = acf.arma11[2:11]
> c1 = acf.arma11
> c2 = pacf.arma11
> arma11 = cbind(c1, c2)
> arma11 # Nilai-nilai ACF dan PACF teoritis
> par(mfrow=c(1,2))
> plot(acf. arma11, type="h", xlab="lag", ylim=c(-1,1))
> abline(h=0)
> plot(pacf. arma11, type="h", xlab="lag", ylim=c(-1,1))
> abline(h=0)

```

Hasil plot ACF dan PACF teoritis dari model ARIMA(1,0,1) yang disingkat model ARMA(1,1), dengan nilai koefisien parameter AR (ϕ) 0,7 dan koefisien MA (θ) 0,4.



Gambar 11.11. Plot ACF dan PACF teoritis model ARMA(1,1) dengan ϕ 0,7 dan θ 0,4

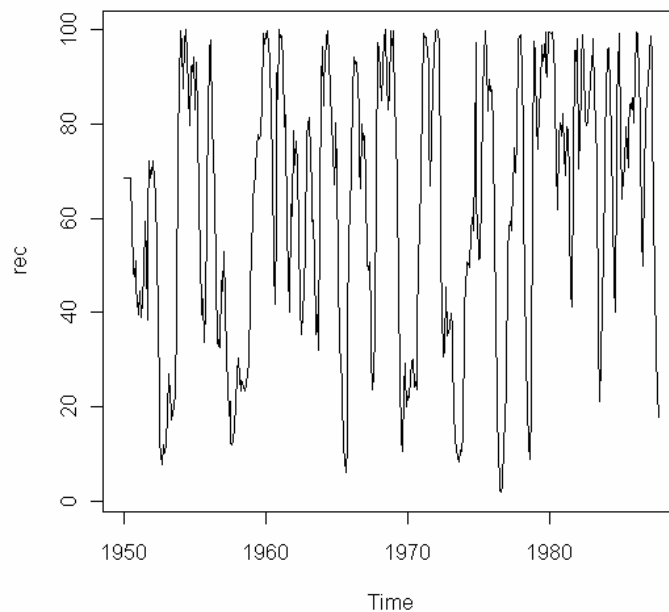
Gambar diatas menunjukkan bahwa plot ACF pada model ARMA(1,1) dengan koefisien parameter ϕ 0,7 dan θ 0,4 adalah *dies down* (turun cepat secara eksponensial). Pola yang sama juga ditunjukkan oleh plot PACF yaitu *dies down* (turun cepat secara sinusoidal) dengan nilai PACF yang berubah dari positif ke negatif seperti petunjuk pada Tabel 11.1 diatas.

11.3.1. Contoh Kasus Model ARIMA Non-musiman yang Stasioner

Misalkan akan dilakukan peramalan dengan model ARIMA pada data runtun waktu **recruit.dat**, yaitu data tentang banyaknya ikan baru yang telah dikumpulkan oleh Dr. Roy Mendelsohn dari *The Pacific Environmental Fisheries Group* (lihat buku Shumway dan Stoffer (2006), dengan judul *"Time Series Analysis and Its Applications with R Examples"*; halaman 7). Data ini adalah data bulanan mulai tahun 1950-1987. Berikut ini adalah **script** file R untuk memanggil dan menampilkan plot time series dari data.

```
> rec = ts(scan("recruit.dat"), start=1950, frequency=12)
> plot(rec)
```

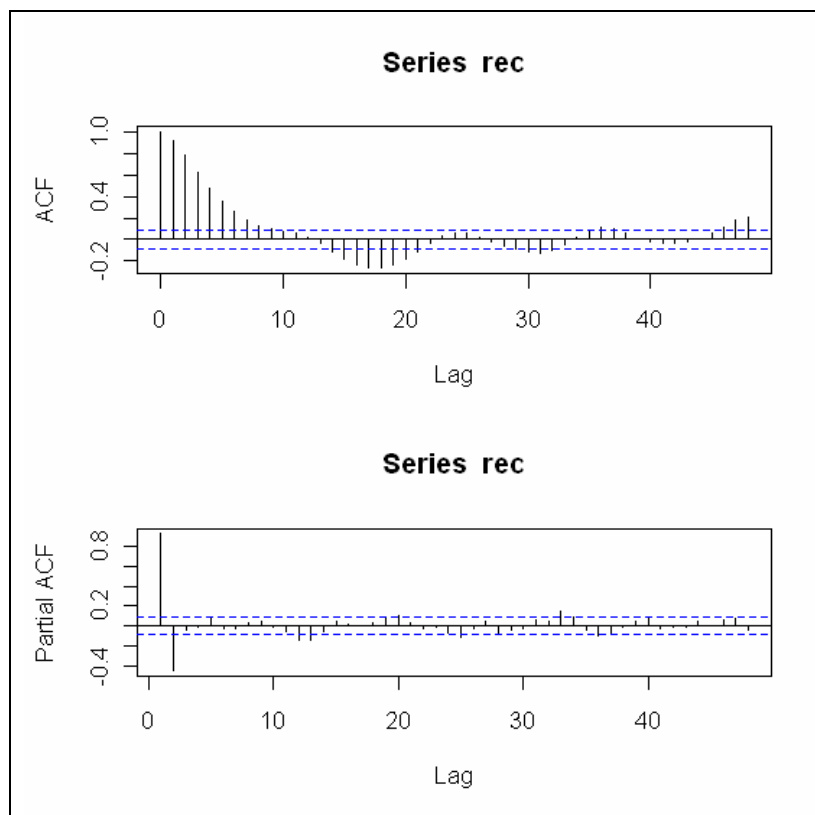
Hasil plot time series dari data adalah sebagai berikut.



Gambar 11.12. Plot time series dari data **recruit**

Dari plot time series pada Gambar 11.12 dapat dijelaskan bahwa data relatif stasioner dan tidak mengandung tren. Berikut adalah **script** file **R** untuk menampilkan ACF dan PACF dari data **recruit** beserta outputnya.

```
> rec = ts(scan("recruit.dat"))  
> plot(rec) # menampilkan plot time series dari data  
> win.graph  
> par(mfrow=c(2,1))  
> acf(rec, 48) # menampilkan ACF sampai lag 48  
> pacf(rec, 48) # menampilkan PACF sampai lag 48
```



Gambar 11.13. Plot ACF dan PACF dari data **recruit**

Hasil identifikasi bentuk ACF dan PACF dari data **recruit** menunjukkan bahwa ACF cenderung *dies down* (turun cepat) dan PACF cenderung terputus setelah lag 2. Dengan demikian, model dugaan yang sesuai untuk data ini adalah ARIMA(2,0,0) atau AR(2). Tahap selanjutnya adalah estimasi parameter pada model ARIMA dugaan. Ada beberapa metode estimasi parameter yang disediakan **R**, antara lain:

- Estimasi *Yule-Walker* (YW)
- Estimasi *Ordinary Least Squares* (OLS)
- Estimasi *Maximum Likelihood Estimation* (MLE)

Berikut ini adalah **script** file **R** untuk estimasi parameter dengan menggunakan metode estimasi Yule-Walker beserta outputnya.

```
> rec.yw = ar.yw(rec, order=2)
> rec.yw$x.mean
[1] 62.26278 # taksiran nilai rata-rata data
> rec.yw$ar
[1] 1.3315874 -0.4445447 # taksiran nilai phi1 dan phi2
> sqrt(diag(rec.yw$sasy.var.coef))
[1] 0.04222637 0.04222637 # standar error dari phi1 dan phi2
> rec.yw$var.pred
[1] 94.79912 # taksiran varians error (mse)
```

Berdasarkan hasil taksiran Yule-Walker pada output diatas, maka model AR(2) yang diperoleh dapat ditulis secara matematis seperti berikut (dua angka belakang koma)

$$(1 - 1.33B + 0.44B^2)(Z_t - 62.26) = a_t,$$

atau

$$Z_t = 62.26(1 - 1.33 + 0.44) + 1.33Z_{t-1} - 0.44Z_{t-2} + a_t,$$

dengan Z_t adalah data asli pada waktu ke- t . Output diatas juga memberikan nilai-nilai standar error pada masing-masing koefisien yang dapat digunakan untuk uji signifikansi parameter-parameter model tersebut.

Sebagai perbandingan, berikut ini adalah **script** file **R** untuk estimasi parameter dengan menggunakan metode estimasi *Ordinary Least Squares* dan *Maximum Likelihood Estimation*, serta output yang dihasilkan.

```

> rec.ols = ar.ols(rec, order=2) # metode OLS
> rec.ols$x.mean
[1] 62.26278 # taksiran nilai rata-rata data
> rec.ols$ar
,, 1
[1]
[1,] 1.3540685 # taksiran nilai phi1 dan phi2
[2,] -0.4631784
> rec.ols$asy.se.coef
$x.mean
[1] 0.4460397
$ar
[1] 0.04178901 0.04187942 # standar error dari phi1 dan phi2
> rec.ols$var.pred
[1]
[1,] 89.71705 # taksiran varians error (mse)

> rec.mle = ar.mle(rec, order=2) # metode MLE
> rec.mle$x.mean
[1] 62.26153 # taksiran nilai rata-rata data
> rec.mle$ar
[1] 1.3512809 -0.4612736 # taksiran nilai phi1 dan phi2
> sqrt(diag(rec.mle$asy.var.coef))
[1] 0.04099159 0.04099159 # standar error dari phi1 dan phi2
> rec.mle$var.pred
[1] 89.33597 # taksiran varians error (mse)

```

Hasil estimasi ketiga metode tersebut menunjukkan bahwa nilai-nilai taksiran parameter model AR(2) yang diperoleh relatif tidak berbeda jauh. Hal yang menarik adalah taksiran dari varians error (atau yang dikenal dengan MSE). Nilai MSE yang diperoleh ketiga model menunjukkan bahwa metode MLE memberikan nilai MSE yang paling kecil, yaitu 89.33597.

Selain menggunakan perintah-perintah diatas, **R** juga menyediakan perintah **arima** untuk estimasi secara langsung dengan menampilkan beberapa nilai taksiran sekaligus. Berikut adalah keterangan penggunaan **arima** dan argumen yang dibutuhkan.

```

arima(x, order = c(0, 0, 0),
      seasonal = list(order = c(0, 0, 0), period = NA),
      xreg = NULL, include.mean = TRUE,
      transform.pars = TRUE,
      fixed = NULL, init = NULL,
      method = c("CSS-ML", "ML", "CSS"),
      n.cond, optim.control = list(), kappa = 1e6)

```

Sebagai contoh, **script** untuk estimasi CSS-MLE (pilihan **default**) pada data **recruit** dengan perintah **arima** beserta outputnya adalah sebagai berikut.

```

> fit1 <- arima(rec, c(2, 0, 0))
> fit1

Call:
arima(x = rec, order = c(2, 0, 0))

Coefficients:
      ar1      ar2  intercept
    1.3512 -0.4612  61.8585
s.e. 0.0416  0.0417   4.0039

sigma^2 estimated as 89.33: log likelihood = -1661.51, aic = 3331.02

```

Hasil diatas menunjukkan bahwa nilai-nilai taksiran parameter model AR(2) untuk data **recruit** adalah sama dengan menggunakan perintah sebelumnya.

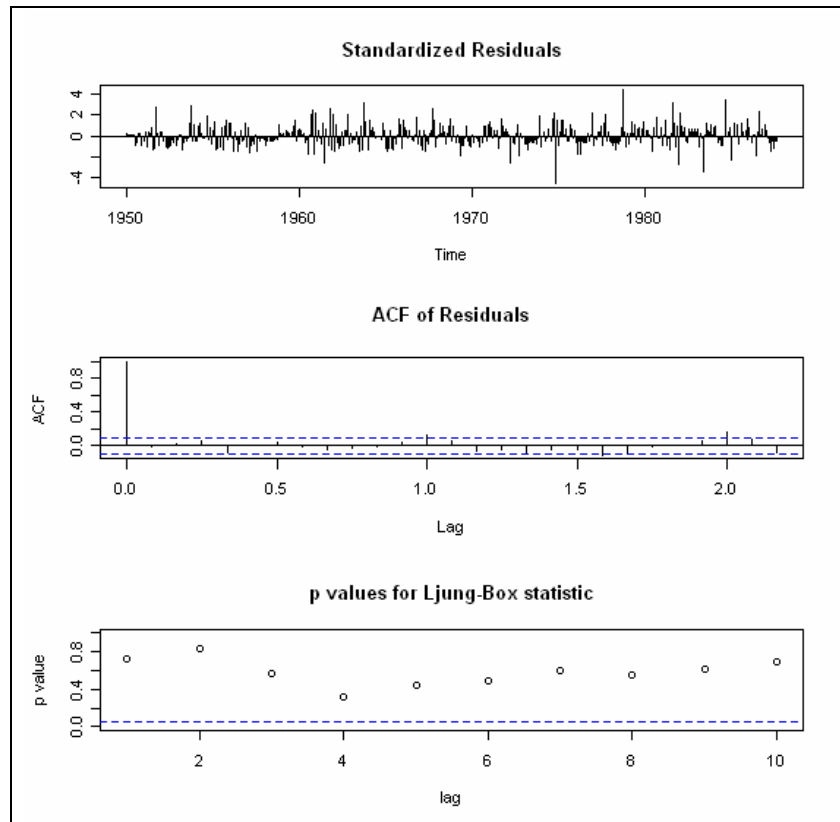
Langkah selanjutnya setelah estimasi parameter diperoleh adalah cek diagnosa untuk mengetahui apakah model sudah memenuhi syarat kebaikan suatu model. **R** menyediakan fasilitas untuk uji kesesuaian model, yaitu Uji Statistik Ljung-Box untuk mengetahui apakah residual model sudah memenuhi syarat white noise. Hal ini dapat dilakukan dengan perintah **tsdiag** seperti contoh berikut ini.

```

> fit1 <- arima(rec, c(2, 0, 0))
> tsdiag(fit1) # cek diagnosa dengan Uji Ljung-Box

```

Berikut adalah output hasil perintah **tsdiag** untuk pengecekan apakah residual model sudah memenuhi syarat white noise.



Gambar 11.14. Plot ACF residual dan p-value dari uji Statistik Ljung-Box

Hasil diatas menunjukkan bahwa residual model AR(2) telah memenuhi syarat white noise. Hal ini ditunjukkan oleh p-value dari uji Ljung-Box yang semuanya lebih besar dari 0,05 (alpha atau tingkat signifikansi pengujian).

Asumsi kedua yang juga harus diperiksa adalah normalitas dari residual model. **R** menyediakan banyak perintah untuk uji normalitas, baik secara grafik atau statistik inferensia. Pada bagian ini akan digunakan histogram dan QQ-plot untuk evaluasi secara grafik. Secara inferensi digunakan salah satu perintah yang ada, yaitu **shapiro.test** untuk menerapkan uji **Shapiro-Wilk**. Berikut adalah **script** file **R** yang dapat digunakan untuk menampilkan histogram dan QQ-plot dari residual, serta uji normalitas residual model dengan uji **Shapiro-Wilk**.

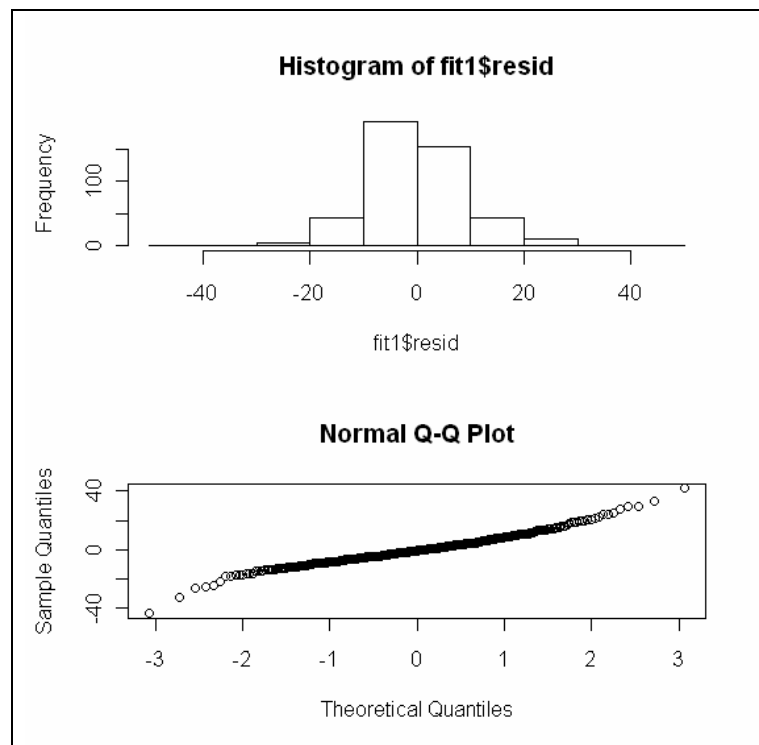

```
> par(mfrow=c(2,1))  
> hist(fit1$resid,br=12)  
> qqnorm(fit1$resid)  
> shapiro.test(fit1$resid)
```

Shapiro-Wilk normality test

data: fit1\$resid

W = 0.9736, p-value = 2.723e-07

Output histogram dan QQ-plot residual model AR(2) pada data **recruit** dapat dilihat pada Gambar 11.15. Output uji Shapiro-Wilk diatas menunjukkan bahwa residual belum memenuhi syarat distribusi normal. Hal ini ditunjukkan oleh p-value yang lebih kecil dari 0.05 (alpha pengujian).



Gambar 11.15. Histogram dan QQ-Plot residual model AR(2) pada data **recruit**

Langkah terakhir setelah model yang diperoleh sudah memenuhi syarat model adalah peramalan. Anggap model AR(2) adalah model yang sesuai untuk peramalan data **recruit**. Berikut ini adalah **script** file **R** yang lengkap mulai tahap identifikasi, estimasi, cek diagnosa, dan peramalan.

```
> rec = ts(scan("recruit.dat"), start=1950, frequency=12)

> plot(rec)                                # Tahap IDENTIFIKASI
> win.graph
> par(mfrow=c(2,1))
> acf(rec, 48) # menampilkan ACF sampai lag 48
> pacf(rec, 48) # menampilkan PACF sampai lag 48

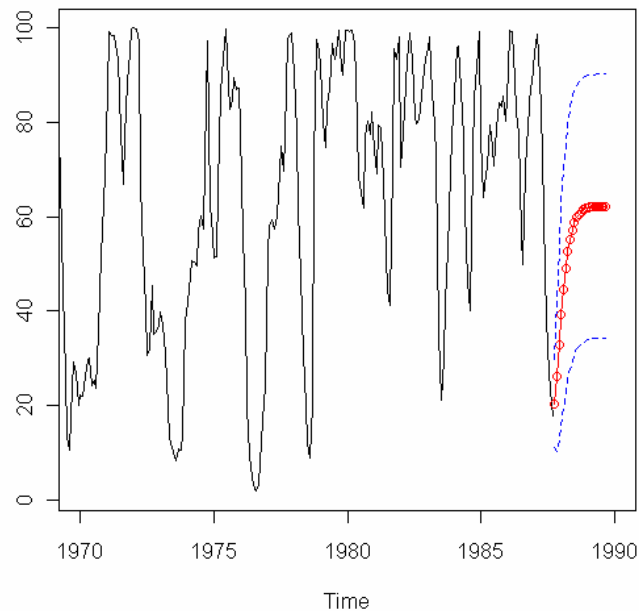
> fit1 <- arima(rec, c(2, 0, 0))            # Tahap ESTIMASI

> tsdiag(fit1)                             # Tahap CEK DIAGNOSA
> win.graph
> par(mfrow=c(2,1))
> hist(fit1$resid, br=12)
> qqnorm(fit1$resid)
> shapiro.test(fit1$resid)

> rec.fore = predict(fit1, n.ahead=24) # Tahap PERAMALAN
> U = rec.fore$pred + rec.fore$se
> L = rec.fore$pred - rec.fore$se
> minx = min(rec, L)
> maxx = max(rec, U)
> ts.plot(rec, rec.fore$pred, xlim=c(1970,1990), ylim=c(minx,maxx))
> lines(rec.fore$pred, col="red", type="o")
> lines(U, col="blue", lty="dashed")
> lines(L, col="blue", lty="dashed")
```

Pada **script** ini nilai-nilai ramalan 24 periode yang akan datang disimpan dalam **object** yang diberi nama **rec.fore**. Dalam hal ini nilai-nilai ramalan beserta batas atas dan batas bawah juga diberikan. Untuk menampilkan angka-angka tersebut cukup dengan menuliskan nama **object** tersebut pada **R-Console**.

Hasil plot ramalan beserta batas atas dan batas bawah ramalan ditampilkan pada Gambar 11.16. Warna merah menunjukkan nilai-nilai ramalan, sedangkan warna biru adalah batas bawah dan atas dari ramalan. Hasil tersebut menunjukkan bahwa ramalan yang diperoleh relatif cukup baik, karena sudah mengikuti pola data **recruit** pada waktu-waktu sebelumnya.



Gambar 11.16. Plot ramalan, batas atas dan batas bawah pada data **recruit**

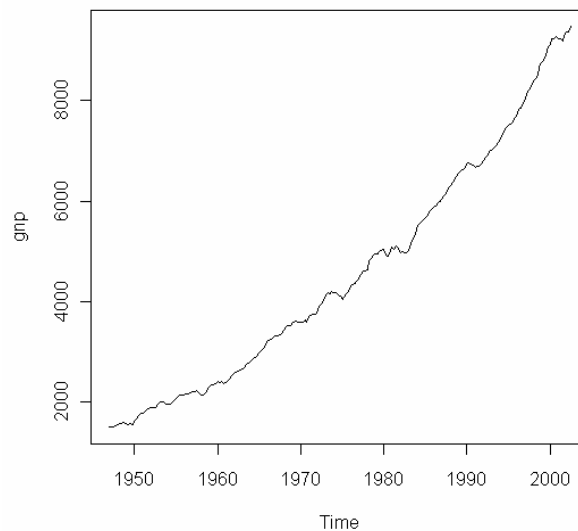
11.3.2. Contoh Kasus Model ARIMA Non-musiman yang Tidak Stasioner

Misalkan akan dilakukan peramalan dengan model ARIMA pada data runtun waktu **gnp96.dat**, yaitu data kuartalan tentang GNP US periode 1947(1) sampai 2002(3). Data ini adalah data runtun waktu yang tidak stasioner dalam mean dan varians, seperti yang terlihat pada Gambar 11.17. Hal ini ditunjukkan oleh fluktuasi varians yang cenderung meningkat seiring bertambahnya waktu. Karena data belum stasioner dalam mean dan varians, maka pada tahap identifikasi dilakukan proses transformasi terlebih dulu untuk menstabilkan varians, dan kemudian differencing untuk menstasionerkan mean data. Pemilihan transformasi yang sesuai dapat menggunakan transformasi **Box-Cox**. Dalam kasus ini, transformasi **log** yang terpilih untuk menstabilkan variansi data. **R** menyediakan perintah **diff** untuk proses differencing suatu data runtun waktu.

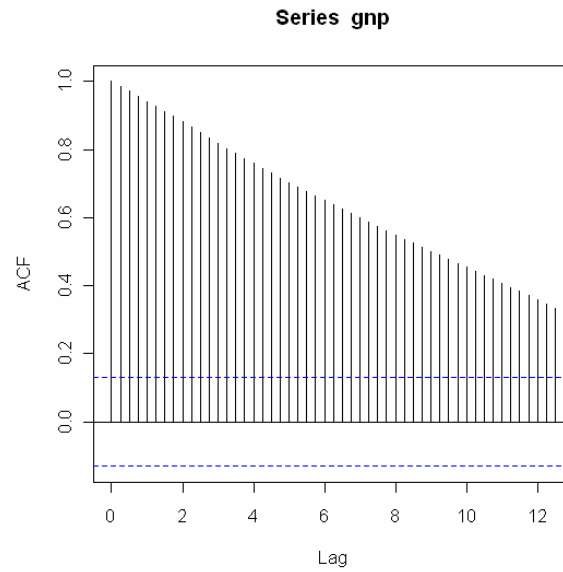
Berikut ini adalah **script** lengkap untuk memanggil data, identifikasi, estimasi, cek diagnosa, dan peramalan pada data **gnp96.dat**. Output lengkap dari **script** ini ditampilkan per tahapan pembentukan model ARIMA dengan menggunakan prosedur Box-Jenkins.

```
> gnp96 = read.table("gnp96.dat")
> gnp = ts(gnp96[,2], start=1947, frequency=4)
> # tahap IDENTIFIKASI
> plot(gnp)
> acf(gnp, 50)
> gnpgr = diff(log(gnp)) # transformasi dan differencing data
> plot.ts(gnpgr)
> par(mfrow=c(2,1))
> acf(gnpgr, 24)
> pacf(gnpgr, 24)
> # tahap ESTIMASI
> gnpgr.ar = arima(gnpgr, order = c(1, 0, 0)) # potential problem here
> gnpgr.ma = arima(gnpgr, order = c(0, 0, 2))
> gnpgr.ar
> gnpgr.ma
> # tahap CEK DIAGNOSA
> tsdiag(gnpgr.ar, gof.lag=20)
> tsdiag(gnpgr.ma, gof.lag=20)
```

▪ Hasil Tahap IDENTIFIKASI

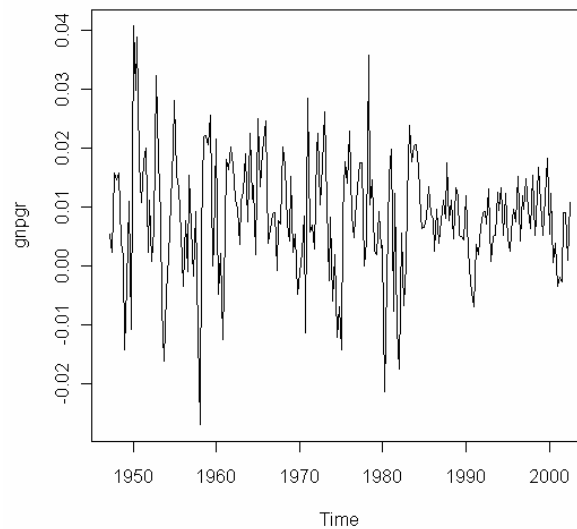


Gambar 11.17. Plot data kuartalan GNP US mulai 1947(1) sampai 2002(3)



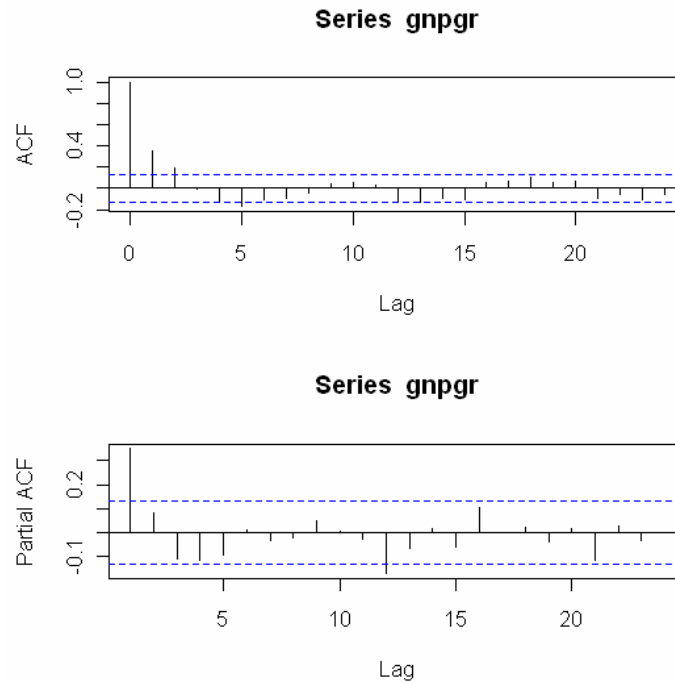
Gambar 11.18. Plot ACF data kuartalan GNP US mulai 1947(1) – 2002(3)

Berikut ini adalah hasil transformasi **log** dan **differencing** untuk mendapatkan data runtun waktu yang stasioner.



Gambar 11.19. Plot data kuartalan GNP US setelah di **log** dan **differencing** (gnpgr)

Output bentuk ACF dan PACF dari data gnp yang sudah ditransformasi dan differencing menjadi data **gnpgr**.



Gambar 11.20. Plot ACF dan PACF dari data **gnpgr**

▪ Hasil Tahap ESTIMASI

```
> gnpgr.ar # Hasil estimasi parameter model ARIMA(1,1,0)

Call:
arima(x = gnpgr, order = c(1, 0, 0))

Coefficients:
    ar1 intercept 
 0.3467  0.0083 
s.e. 0.0627  0.0010 

sigma^2 estimated as 9.03e-05: log likelihood = 718.61, aic = -1431.22
```

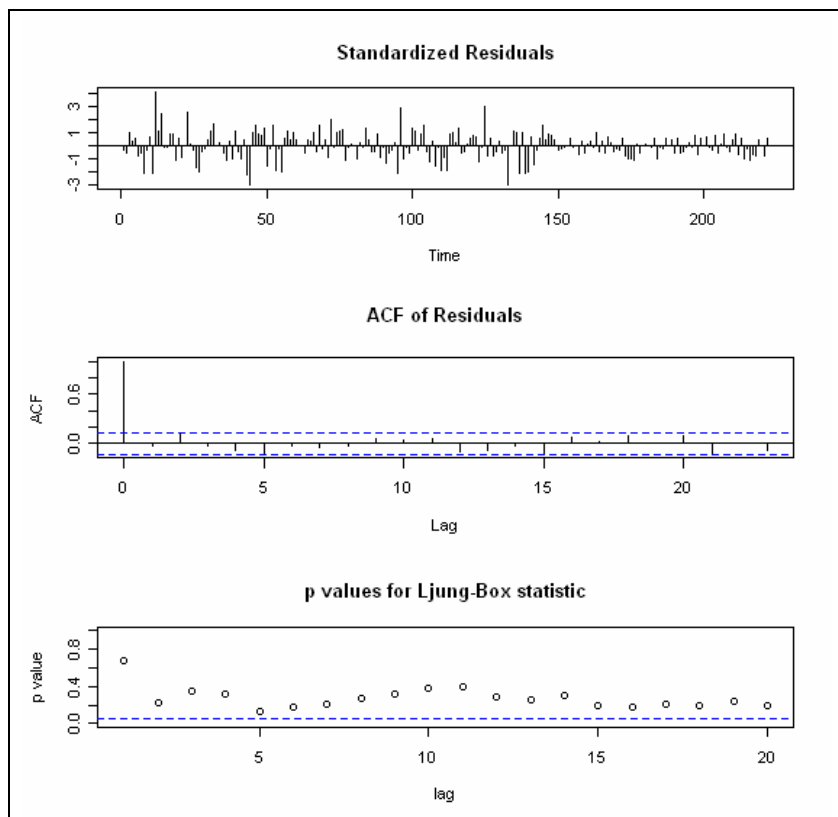
```
> gnpgr.ma # Hasil estimasi parameter model ARIMA(0,1,2)

Call:
arima(x = gnpgr, order = c(0, 0, 2))

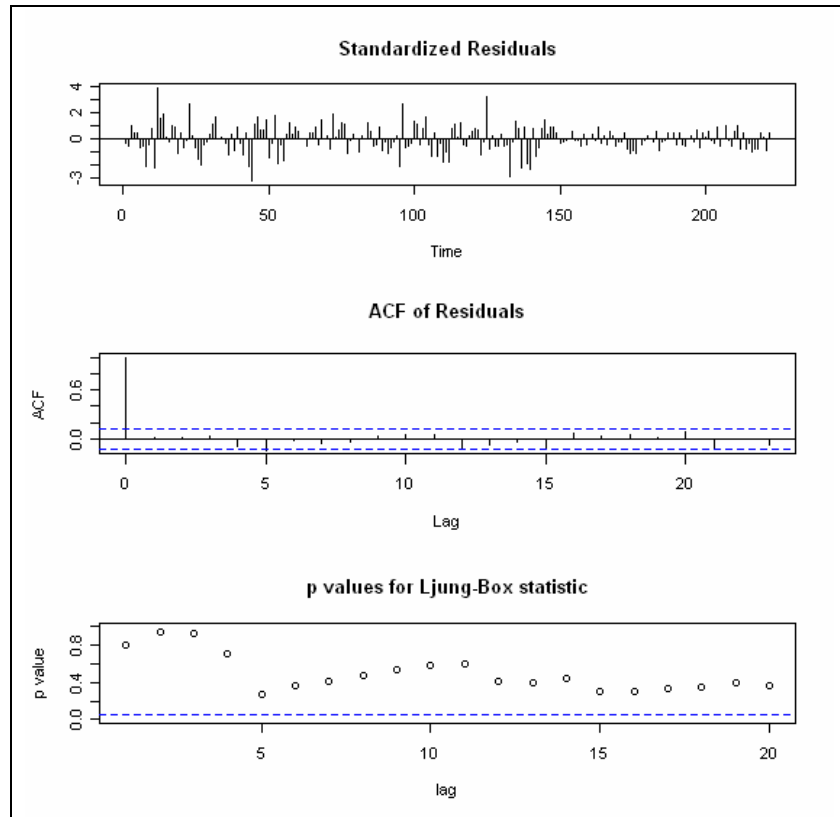
Coefficients:
      ma1      ma2  intercept 
 0.3028  0.2035   0.0083 
s.e. 0.0654  0.0644   0.0010 

sigma^2 estimated as 8.92e-05: log likelihood = 719.96, aic = -1431.93
```

▪ Hasil Tahap CEK DIAGNOSA



Gambar 11.20. Plot ACF dan PACF dari residual model ARIMA(1,1,0)



Gambar 11.21. Plot ACF dan PACF dari residual model ARIMA(0,1,2)

Hasil-hasil diatas menunjukkan bahwa model ARIMA(0,1,2) adalah model ARIMA yang lebih sesuai untuk data GNP US jika dibandingkan dengan model ARIMA(1,1,0). Hal ini ditunjukkan oleh nilai likelihood yang lebih besar dan nilai AIC yang lebih kecil pada model ARIMA(0,1,2).

11.3.3. Model ARIMA Musiman

Secara umum, model ARIMA musiman terdiri dari dua macam yaitu model musiman saja atau $ARIMA(P,D,Q)^S$ dan model ARIMA multiplikatif musiman dan nonmusiman atau $ARIMA(p,d,q)(P,D,Q)^S$, dengan S adalah periode musiman. Bentuk matematis dari model $ARIMA(P,D,Q)^S$ dapat ditulis sebagai berikut

$$(1 - \Phi_{1S}B - \dots - \Phi_P B^{PS})(1 - B^S)^D Z_t = \theta_0 + (1 - \Theta_1 B - \dots - \Theta_Q B^{QS})a_t.$$

Seperti pada model nonmusiman, penentuan orde P dan Q dari model ARIMA musiman pada suatu data runtun waktu dilakukan dengan mengidentifikasi plot ACF dan PACF dari data yang sudah stasioner. Berikut ini adalah petunjuk umum untuk penentuan orde P dan Q pada suatu data runtun waktu musiman yang sudah stasioner.

Tabel 11.2. Pola teoritis ACF dan PACF dari proses musiman yang stasioner

Proses	ACF	PACF
$AR(P)^S$	<i>Dies down</i> pada lag kS , dengan $k=1,2,3,\dots$	<i>Cuts off</i> setelah lag PS
$MA(Q)^S$	<i>Cuts off</i> setelah lag QS	<i>Dies down</i> pada lag kS , dengan $k=1,2,3,\dots$
$ARMA(P,Q)^S$	<i>Dies down</i> pada lag kS , dengan $k=1,2,3,\dots$	<i>Dies down</i> pada lag kS , dengan $k=1,2,3,\dots$
$AR(P)^S$ atau $MA(Q)^S$	<i>Cuts off</i> setelah lag QS	<i>Cuts off</i> setelah lag PS
<i>White noise</i> (Random)	Tidak ada yang signifikan (tidak ada yang keluar batas)	Tidak ada yang signifikan (tidak ada yang keluar batas)

Selanjutnya, gabungan petunjuk pola ACF dan PACF pada Tabel 11.1 dan 11.2 dapat digunakan untuk menentukan orde p , q , P , dan Q pada model musiman multiplikatif $ARIMA(p,d,q)(P,D,Q)^S$. Secara umum bentuk model ARIMA Box-Jenkins Musiman atau $ARIMA(p,d,q)(P,D,Q)^S$ adalah : (Cryer, 1986; Wei, 2006)

$$\phi_p(B)\phi_P(B^S)(1-B)^d(1-B^S)^D Z_t = \theta_0 + \theta_q(B)\theta_Q(B^S)a_t,$$

dengan

p, d, q = order **AR, MA** dan **differencing** Non-musiman,

P, D, Q = order **AR, MA** dan **Differencing** Musiman,

$$\phi_p(B) = (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p),$$

$$\begin{aligned}
\phi_P(B^S) &= (1 - \phi_1 B^S - \phi_2 B^{2S} - \dots - \phi_P B^{PS}), \\
(1 - B)^d &= \text{operasi matematis dari **differencing** Non-musiman,} \\
(1 - B^S)^D &= \text{operasi matematis dari **differencing** Musiman,} \\
\theta_q(B) &= (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q), \\
\theta_Q(B^S) &= (1 - \theta_1 B^S - \theta_2 B^{2S} - \dots - \theta_Q B^{QS}), \\
Z_t &= Z_t - \mu.
\end{aligned}$$

Seperti pada model nonmusiman, **R** menyediakan fasilitas untuk membuat bentuk ACF dan PACF teoritis dari model-model ARIMA musiman yang stasioner, baik musiman saja atau multiplikatif musiman dan nonmusiman. Berikut ini adalah contoh **script** untuk membuat plot ACF dan PACF teoritis dari model ARIMA(p,d,q)(P,D,Q)⁵.

```

> # ACF dan PACF teoritis untuk MA(1)(1)12
> theta = c(-0.6, rep(0,10), -0.5, 0.3)
> # phi = c(rep(0,11), 0.8) untuk model AR
> acf.arma = ARMAacf(ar=0, ma=theta, 60)
> pacf.arma = ARMAacf(ar=0, ma=theta, 60, pacf=T)
> acf.arma = acf.arma[2:61]
> c1 = acf.arma
> c2 = pacf.arma
> arma = cbind(c1, c2)
> arma # Nilai-nilai ACF dan PACF teoritis
> par(mfrow=c(1,2))
> plot(acf.arma, type="h", xlab="lag", ylim=c(-1,1))
> abline(h=0)
> plot(pacf.arma, type="h", xlab="lag", ylim=c(-1,1))
> abline(h=0)

```

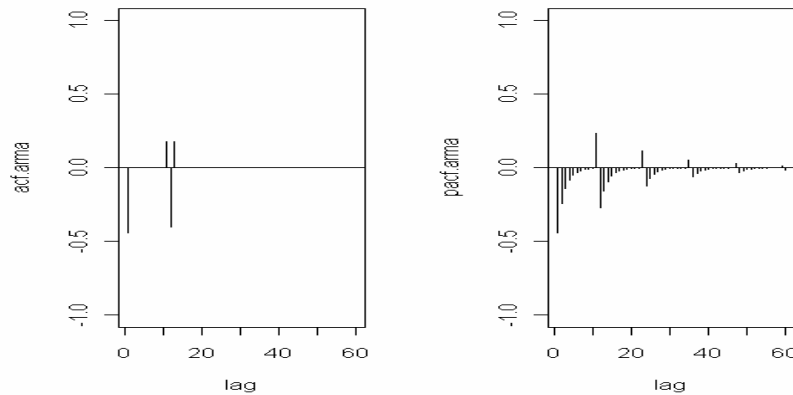
Model ARIMA yang digunakan pada **script** diatas adalah ARIMA(0,0,1)(0,0,1)¹² atau disingkat MA(1)(1)¹². Secara matematis model ini dapat ditulis dalam bentuk

$$Z_t = (1 - \theta_1 B)(1 - \Theta_1 B^{12})a_t,$$

atau

$$Z_t = a_t - \theta_1 a_{t-12} - \Theta_1 a_{t-12} + \theta_1 \Theta_1 a_{t-13}.$$

Hasil plot ACF dan PACF teoritis dari model ARIMA(0,0,1)(0,0,1)¹² dengan nilai koefisien parameter model (*tetha1* dan *TETHA1*) 0,6 dan 0,5 dapat dilihat pada Gambar 11.22 berikut ini.



Gambar 11.22. Plot ACF dan PACF teoritis model ARIMA(0,0,1)(0,0,1)¹²

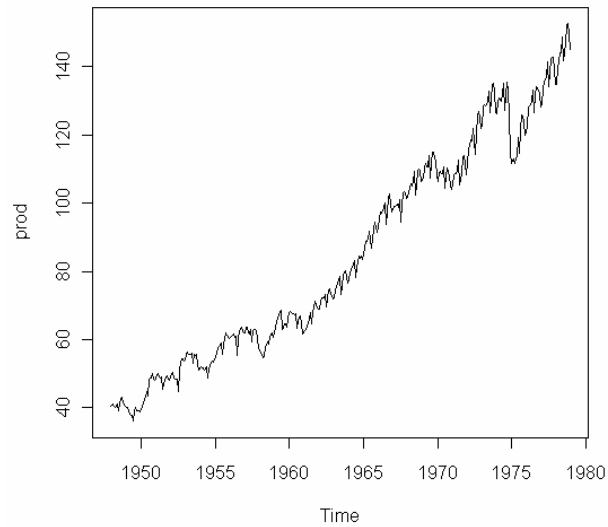
11.3.4. Contoh Kasus Model ARIMA Musiman

Misalkan akan dilakukan peramalan dengan model ARIMA pada data runtun waktu **prod.dat**, yaitu data bulanan tentang *the Federal Reserve Board Procuction Index* (lihat buku Shumway dan Stoffer (2006), dengan judul “*Time Series Analysis and Its Applications with R Examples*”; halaman 160). Data ini adalah data bulanan mulai tahun 1948-1978. Berikut ini adalah **script** file R untuk memanggil dan menampilkan plot time series, ACF, dan PACF dari data asli.

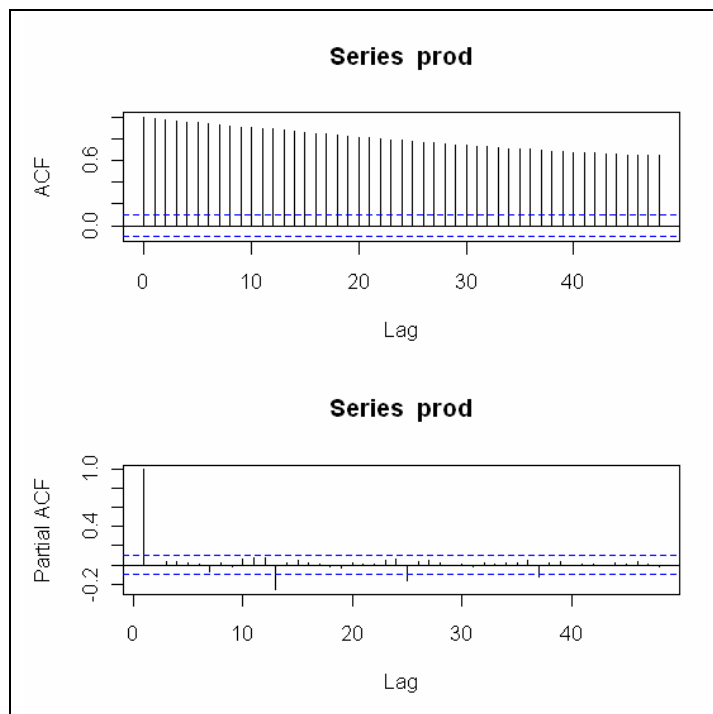
```
> prod=ts(scan("prod.dat"), start=1948, frequency=12)
> plot(prod)
> par(mfrow=c(2,1)) # (P)ACF of data
> acf(prod, 48)
> pacf(prod, 48)
```

Hasil dari plot time series data indeks produksi tersebut (prod) dapat dilihat pada Gambar 11.23. Plot ini menunjukkan bahwa data mengandung tren naik atau data belum stasioner dalam mean.

Hal ini didukung oleh bentuk ACF dan PACF data asli pada Gambar 11.24, khususnya pola ACF yang turun lambat yang mengindikasikan bahwa data belum stasioner dalam mean. Pada tahap ini (tahap identifikasi) dilakukan **differencing** pada data untuk mendapatkan data yang stasioner dalam mean.



Gambar 11.23. Plot time series dari data indeks produksi (**prod**)

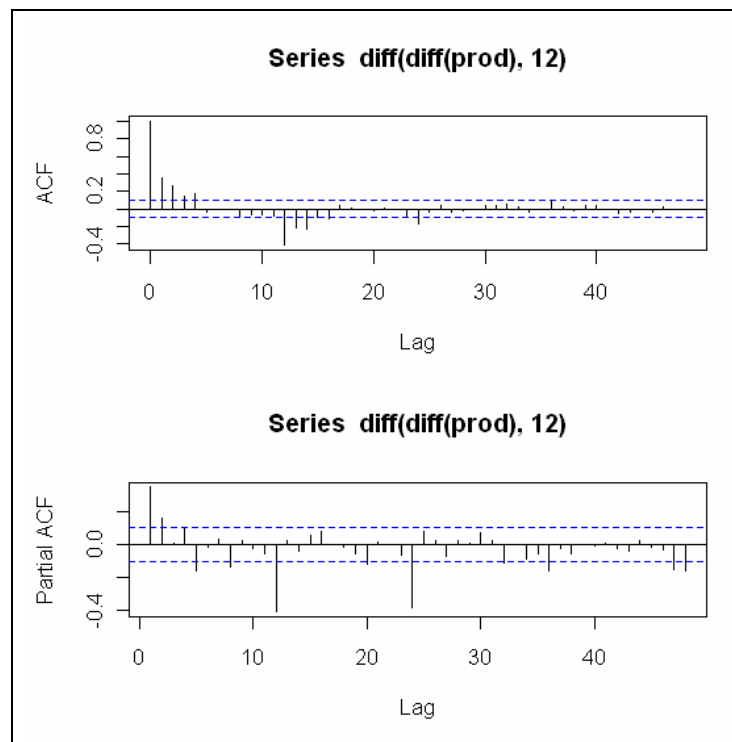


Gambar 11.24. Plot ACF dan PACF dari data indeks produksi (**prod**)

Berikut ini adalah **script** untuk **differencing** dan identifikasi bentuk ACF dan PACF dari data yang sudah stasioner.

```
> par(mfrow=c(2,1)) # ACF dan PACF differencing d=1
> acf(diff(prod), 48)
> pacf(diff(prod), 48)
> # Karena belum stasioner dalam musiman, dilanjutkan
> # differencing musiman D=1, S=12 atau D=12.
> par(mfrow=c(2,1)) # ACF dan PACF differencing d=1, D=12
> acf(diff(diff(prod),12), 48)
> pacf(diff(diff(prod),12), 48)
```

Hasil dari plot ACF dan PACF pada data yang sudah **didifferencing** $d=1$, dan $D=1$, $S=12$, atau **differencing** nonmusiman dan musiman, ditampilkan pada Gambar 11.25.



Gambar 11.25. Plot ACF dan PACF dari data **prod** yang telah didifferencing

Gambar ACF dan PACF data yang sudah stasioner menunjukkan bahwa pada lag-lag nonmusiman (lag 1-9) ACF dan PACF cenderung *dies down*. Hal ini juga terjadi pada lag-lag musiman (lag 12, 24, 36) yang cenderung juga *dies down*. Berdasarkan petunjuk pada Tabel 9.1 dan 9.2, diduga ada 3 (tiga) model yang sesuai untuk data ini, yaitu:

1. ARIMA(1,1,1)(0,1,1)¹²
2. ARIMA(1,1,1)(2,1,0)¹²
3. ARIMA(1,1,1)(2,1,1)¹².

Hasil estimasi pada ketiga model dugaan tersebut menunjukkan bahwa model ARIMA(1,1,1)(2,1,1)¹² merupakan model terbaik, berdasarkan perbandingan kriteria AIC. Berikut adalah **script R** yang dapat digunakan untuk estimasi parameter pada model ARIMA multiplikatif.

```
> # Tahap ESTIMASI PARAMETER model ke-3
> prod.fit3 = arima(prod, order=c(1,1,1), seasonal=list(order=c(2,1,1), period=12))
> prod.fit3
```

Berikut ini adalah output hasil estimasi parameter dari model ketiga, yaitu model multiplikatif ARIMA(1,1,1)(2,1,1)¹². Hasil ini menunjukkan bahwa model multiplikatif ARIMA(1,1,1)(2,1,1)¹² adalah sesuai untuk data **prod** khususnya jika dilihat dari taksiran parameter dan signifikansi parameter tersebut (hitung uji statistik **t** nya).

```
> prod.fit3 # Menampilkan hasil-hasil estimasi parameter

Call:
arima(x = prod, order = c(1, 1, 1), seasonal = list(order = c(2, 1, 1), period = 12))

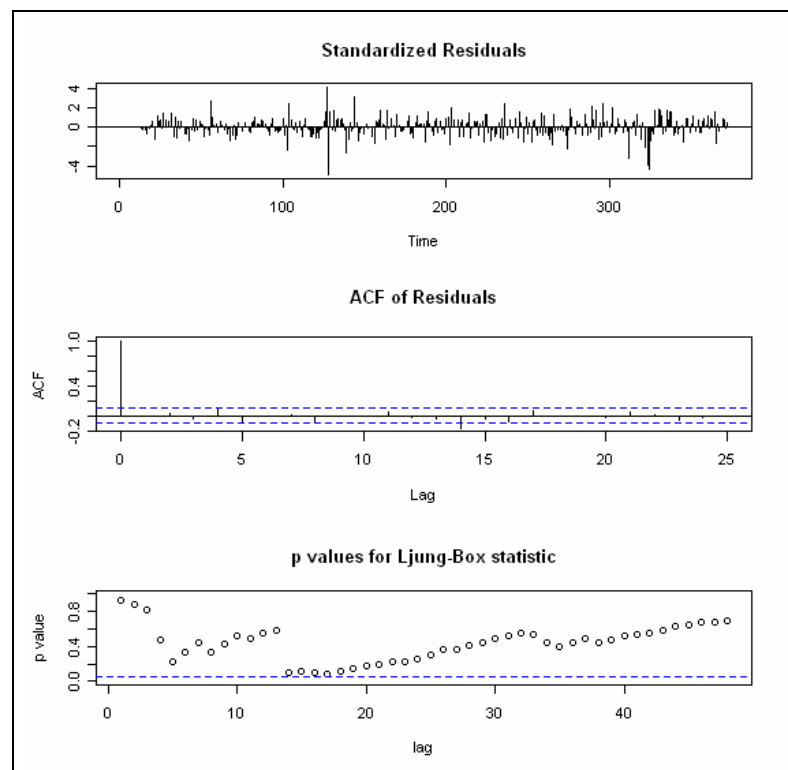
Coefficients:
      ar1      ma1      sar1      sar2      sma1
 0.5753 -0.2709 -0.2153 -0.2800 -0.4968
s.e. 0.1120 0.1300 0.0784 0.0619 0.0712

sigma^2 estimated as 1.351: log likelihood = -568.22, aic = 1148.43
```

Tahap selanjutnya yaitu tahap cek diagnosa dan peramalan berdasarkan model yang sesuai. Berikut adalah **script R** yang dapat digunakan untuk tahap cek diagnosa dan peramalan pada model ARIMA multiplikatif, serta outputnya.

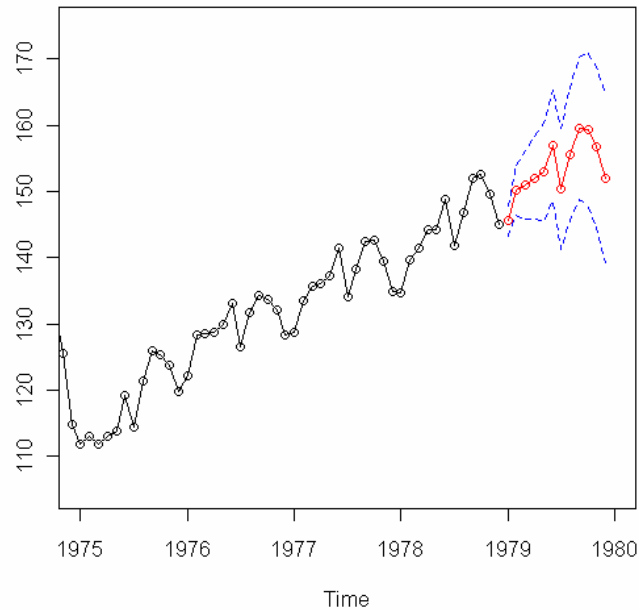
```
> # Tahap DIAGNOSTICS CHECK
> tsdiag(prod.fit3, gof.lag=48)

> # Tahap PERAMALAN
> prod.pr = predict(prod.fit3, n.ahead=12)
> U = prod.pr$pred + 2*prod.pr$se
> L = prod.pr$pred - 2*prod.pr$se
> ts.plot(prod,prod.pr$pred, col=1:2, type="o", ylim=c(105,175), xlim=c(1975,1980))
> lines(U, col="blue", lty="dashed")
> lines(L, col="blue", lty="dashed")
```



Gambar 11.26. Output tahap cek diagnosa pada model $ARIMA(1,1,1)(2,1,1)^{12}$

Plot perbandingan antara nilai aktual dan nilai ramalan beberapa periode kedepan dapat dilihat pada gambar berikut ini.



Gambar 11.27. Plot ramalan, batas atas dan batas bawah pada data **prod**

Dengan demikian model terbaik yang diperoleh untuk data **prod** diatas adalah model multiplikatif $ARIMA(1,1,1)(2,1,1)^{12}$ yang secara matematis dapat ditulis dalam bentuk sebagai berikut (dua angka belakang koma).

$$(1 - 0,58B)(1 + 0,22B^{12} + 0,28B^{24})(1 - B)(1 - B^{12})Z_t = (1 - 0,27B)(1 - 0,50B^{12})a_t,$$

dengan taksiran varians error (MSE) sebesar 1,351.

Penjabaran dari model $ARIMA(1,1,1)(2,1,1)^{12}$ di atas akan menunjukkan bahwa peramalan indeks produksi pada suatu bulan (Z_t) merupakan fungsi linear dari indeks produksi pada bulan-bulan sebelumnya, yaitu satu (Z_{t-1}), dua (Z_{t-2}), dua belas (Z_{t-12}), tiga belas (Z_{t-13}), empat belas (Z_{t-14}), dua puluh empat (Z_{t-24}), dua puluh lima (Z_{t-25}), dua puluh enam (Z_{t-26}), tiga puluh enam (Z_{t-36}), tiga puluh tujuh (Z_{t-37}), tiga puluh delapan (Z_{t-38}), dan residual pada bulan-bulan sebelumnya, yaitu satu (a_{t-1}), dua belas (a_{t-12}), dan tiga belas (a_{t-13}) bulan sebelumnya.

11.3.5. Kriteria Pemilihan Model

Ada beberapa kriteris pemilihan model yang dapat digunakan untuk memilih model ARIMA terbaik pada suatu data runtun waktu, antara lain **Akaike's Information Criterion (AIC)**, **AIC Bias Corrected (AICc)**, dan **Schwarz's Information Criterion (SIC)**. Berikut ini adalah rumus untuk perhitungan kriteria-kriteria tersebut.

- **Akaike's Information Criterion (AIC)**

$$AIC = \ln \hat{\sigma}_k^2 + \frac{n + 2k}{n},$$

dengan k adalah banyaknya parameter dalam model, dan n adalah jumlah data (pengamatan), serta $\hat{\sigma}_k^2$ estimator maksimum likelihood dari varians error yang didefinisikan sebagai berikut

$$\hat{\sigma}_k^2 = \frac{RSS_k}{n},$$

dengan RSS adalah **the residual sum of squares** (jumlah kuadrat error).

- **AIC Bias Corrected (AICc),**

$$AICc = \ln \hat{\sigma}_k^2 + \frac{n + k}{n - k - 2},$$

dengan k , n , dan $\hat{\sigma}_k^2$ seperti yang didefinisikan diatas.

- **Schwarz's Information Criterion (SIC).**

$$SIC = \ln \hat{\sigma}_k^2 + \frac{k \ln n}{n},$$

dengan k , n , dan $\hat{\sigma}_k^2$ seperti yang didefinisikan diatas.

Misalkan akan dilakukan perbandingan nilai-nilai kriteria **AIC**, **AICc**, dan **SIC** untuk pemilihan model terbaik pada kasus data GNP US sebelumnya, yaitu antara model ARIMA(1,1,0) dengan ARIMA(0,1,2). Seperti yang dijelaskan pada sub-bab 11.3.3, kriteria **AIC** menunjukkan bahwa model ARIMA(0,1,2) adalah model yang lebih baik dibanding model ARIMA(1,1,0). Hal ini dikarenakan model ARIMA(0,1,2) memberikan nilai **AIC** yang lebih kecil dibandingkan dengan model ARIMA(1,1,0). Berikut ini adalah **script** dan output untuk perhitungan nilai-nilai kriteria **AIC**, **AICc**, dan **SIC** pada kedua model tersebut.

```

> gnp96 = read.table("gnp96.dat")
> gnp = ts(gnp96[,2], start=1947, frequency=4)
> gnpgr = diff(log(gnp))
> gnpgr.ar = arima(gnpgr, order = c(1, 0, 0))
> gnpgr.ma = arima(gnpgr, order = c(0, 0, 2))
>
> n = length(gnpgr)           # jumlah data
> kma = length(gnpgr.ma$coef) # jumlah parameter pada model MA
> sma=gnpgr.ma$sigma2         # nilai mle dari sigma^2
> kar = length(gnpgr.ar$coef) # jumlah parameter pada model MA
> sar=gnpgr.ar$sigma2         # nilai mle of sigma^2
>
> # Perhitungan nilai AIC
> log(sma) + (n+2*kma)/n      # MA(2)
[1] -8.297695
> log(sar) + (n+2*kar)/n      # AR(1)
[1] -8.294403
>
> # Perhitungan nilai AICc
> log(sma) + (n+kma)/(n-kma-2) # MA(2)
[1] -8.287855
> log(sar) + (n+kar)/(n-kar-2) # AR(1)
[1] -8.284898
>
> # Perhitungan nilai BIC
> log(sma) + kma*log(n)/n      # MA(2)
[1] -9.251712
> log(sar) + kar*log(n)/n      # AR(1)
[1] -9.263748

```

Output di atas menunjukkan bahwa kriteria AICc dan SIC memberikan hasil yang berbeda. Kriteria AICc memberikan hasil yang sama dengan AIC, yaitu model terbaik adalah model ARIMA(0,1,2). Hal ini ditunjukkan oleh nilai AICc pada model ARIMA(0,1,2) yang lebih kecil daripada model ARIMA(1,1,0). Sebaliknya, kriteria SIC menunjukkan bahwa model yang lebih sederhana adalah yang lebih baik, yaitu model ARIMA(1,1,0). Nilai SIC pada model ARIMA(1,1,0) adalah -9,288 dan ini lebih kecil dibanding yang diperoleh model ARIMA(0,1,2), yaitu -9,276. Seringkali dalam banyak kasus, kriteria SIC akan cenderung memilih model yang lebih sederhana dibanding kriteria AIC dan AICc.

11.4. Rangkuman perintah dan library yang berkaitan dengan Analisis Runtun Waktu

Berikut ini adalah rangkuman beberapa **perintah** dan penjelasan tentang kegunaan, serta **library** dari **perintah** tersebut, yang biasanya digunakan dalam analisis runtun waktu.

▪ Input Data Runtun Waktu

Perintah	Kegunaan	library
cycle()	<i>gives the positions in the cycle of each observation</i>	stats
deltat()	<i>returns the time interval between observations</i>	stats
end()	<i>extracts and encodes the times the last observation were taken</i>	stats
frequency()	<i>returns the number of samples per unit time</i>	stats
start()	<i>reads a time series file</i>	stats
time()	<i>extracts and encodes the times the first observation were taken</i>	stats
ts()	<i>creates time-series objects</i>	stats
window()	<i>is a generic function which extracts the subset of the object 'x' observed between the times 'start' and 'end'. If a frequency is specified, the series is then re-sampled at the new frequency</i>	stats

▪ Dekomposisi Runtun Waktu

Perintah	Kegunaan	library
decompose()	<i>decomposes a time series into seasonal, trend and irregular components using moving averages. Deals with additive or multiplicative seasonal component</i>	stats
filter()	<i>linear filtering on a time series</i>	stats
HoltWinters()	<i>computes Holt-Winters Filtering of a given time series</i>	stats

Perintah	Kegunaan	library
sfilter()	<i>removes seasonal fluctuation using a simple moving average</i>	ast
spectrum()	<i>estimates the spectral density of a time series</i>	stats
stl()	<i>decomposes a time series into seasonal, trend and irregular components using 'loess'</i>	stats
tsr()	<i>decomposes a time series into trend, seasonal and irregular. Deals with additive and multiplicative components</i>	ast

▪ **Pengujian dalam Analisis Runtun Waktu**

Perintah	Kegunaan	library
adf.test()	<i>computes the Augmented Dickey-Fuller test for the null that 'x' has a unit root (tseries)</i>	tseries
Box.test()	<i>computes the Box-Pierce or Ljung-Box test statistic for examining the null hypothesis of independence in a given time series</i>	stats
bds.test()	<i>computes and prints the BDS test statistic for the null that 'x' is a series of i.i.d. random variables</i>	tseries
bptest()	<i>performs the Breusch-Pagan test for heteroskedasticity of residuals</i>	lmtest
dwtest()	<i>performs the Durbin-Watson test for autocorrelation of residuals</i>	lmtest
jarque.bera.test()	<i>Jarque-Bera test for normality</i>	tseries
kpss.test()	<i>computes KPSS test for stationarity</i>	tseries
shapiro.test()	<i>Shapiro-Wilk Normality Test</i>	stats
tsdiag()	<i>a generic function to plot time-series diagnostics</i>	stats

▪ **Model-model Stokastik dalam Analisis Runtun Waktu**

Perintah	Kegunaan	library
ar()	<i>fits an autoregressive time series model to the data, by default selecting the complexity by AIC</i>	stats
arima()	<i>fits an ARIMA model to a univariate time series</i>	stats
arima.sim()	<i>simulate from an ARIMA model</i>	stats
arma()	<i>fits an ARMA model to a univariate time series by conditional least squares</i>	tseries
garch()	<i>fits a Generalized Autoregressive Conditional Heteroscedastic GARCH(p, q) time series model to the data by computing the maximum-likelihood estimates of the conditionally normal model</i>	tseries

▪ **Grafik dalam Analisis Runtun Waktu**

Perintah	Kegunaan	library
lag.plot()	<i>plots time series against lagged versions of themselves. Helps visualizing "auto-dependence" even when auto-correlations vanish</i>	stats
plot.ts()	<i>plotting time-series objects</i>	stats
seaplot()	<i>plotting seasonal sub-series or profile</i>	ast
ts.plot()	<i>plots several time series on a common plot. Unlike 'plot.ts' the series can have a different timebases, but they should have the same frequency</i>	stat
ccf(), pacf(), ccf()	<i>the function 'acf' computes (and by default plots) estimates of the autocovariance or autocorrelation function. Function 'pacf' is the function used for the partial autocorrelations. Function 'ccf' computes the cross-correlation or cross-covariance of two univariate series</i>	stats
diff.ts()	<i>returns suitably lagged and iterated differences</i>	stats
lag()	<i>computes a lagged version of a time series, shifting the time base back by a given number of observations</i>	stats

BAB 12

ANALISIS MULTIVARIAT DENGAN R

Analisis Multivariat merupakan salah satu metode dalam analisis statistik yang banyak digunakan dalam penelitian kuantitatif yang melibatkan banyak variabel. Ada beberapa metode dalam Analisis Multivariat, antara lain Analisis Faktor, Analisis Diskriminan, Analisis Klaster, *Multidimensional Scaling*, Analisis Konjoin, dan Model Persamaan Struktural (SEM). Secara lengkap teori yang berkaitan dengan analisis multivariat dapat dilihat pada Johnson dan Wichern (1998), Sharma (1996), serta Hair dkk. (2006). Pada bab ini akan dijelaskan penggunaan **R** untuk beberapa metode tersebut, khususnya Analisis Faktor, Analisis Diskriminan, dan Analisis Klaster.

12.1. Analisis Faktor

Analisis Faktor merupakan salah satu metode interdependensi dalam analisis multivariat yang biasanya digunakan untuk mengeksplorasi struktur hubungan yang terjadi dalam suatu kelompok variabel. Selain itu, Analisis Faktor juga digunakan untuk mereduksi dimensi data kedalam suatu variabel baru yang independen yang disebut dengan faktor atau variabel *latent*. Secara umum, ada dua macam Analisis Faktor yaitu Analisis Faktor eksploratori dan konfirmatori. Pada bagian ini, penjelasan penggunaan **R** hanya difokuskan pada Analisis Faktor eksploratori.

R menyediakan fasilitas untuk Analisis Faktor pada library **stats** dengan perintah **factanal**. Misalkan akan dilakukan Analisis Faktor terhadap variabel-variabel tentang persepsi pelanggan pada data **HBAT.SAV** yang ada di buku Hair dkk. (2006, hal. 28-31) dengan judul **Multivariate Data Analysis** seperti yang juga dibahas di Bab 7. Ada 13 variabel persepsi pelanggan yang akan dievaluasi struktur hubungannya, yaitu x_6, x_7, \dots, x_{18} . Identifikasi kecukupan data menunjukkan bahwa variabel x_{15} dan x_{17} tidak memenuhi syarat kecukupan data sehingga kedua variabel tersebut tidak diikuti dalam Analisis Faktor. Penjelasan tentang hasil identifikasi ini secara lengkap dapat dilihat di Hair dkk. (2006) pada Bab 3 tentang Analisis Faktor.

Perintah **factanal** pada **R** adalah fasilitas untuk Analisis Faktor dengan metode ekstraksi Maksimum Likelihood. Ada beberapa pilihan rotasi dan metode untuk mendapatkan faktor skor. Berikut ini adalah deskripsi penggunaan secara umum dari perintah tersebut.

```
factanal(x, factors, data = NULL, covmat = NULL, n.obs = NA,
subset, na.action, start = NULL,
scores = c("none", "regression", "Bartlett"),
rotation = "varimax", control = NULL, ...)
```

Untuk kasus data persepsi pelanggan pada data **HBAT.SAV** yang sudah diimport ke dalam **R** dengan nama **hbat**, berikut adalah perintah dan output dari Analisis Faktor.

```
> FA <- factanal(~x6+x7+x8+x9+x10+x11+x12+x13+x14+x16+x18,
+               factors=4, data=hbat, rotation="varimax", scores="regression")

> FA

Call:
factanal(x = ~x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13 + x14 + x16 + x18,
  factors = 4, data = hbat, scores = "regression", rotation = "varimax")

Uniquenesses:
      x6      x7      x8      x9     x10     x11     x12     x13     x14     x16     x18
0.682 0.360 0.228 0.178 0.679 0.005 0.017 0.636 0.163 0.347 0.076

Loadings:
      Factor1 Factor2 Factor3 Factor4
x6              0.557
x7              0.793
x8              0.872 0.102
x9 0.884 0.142      0.135
x10 0.190 0.521     -0.110
x11 0.502      0.104 0.856
x12 0.119 0.974     -0.130
x13      0.225 -0.216 -0.514
x14      0.894 0.158
x16 0.794 0.101 0.105
x18 0.928 0.189      0.164

      Factor1 Factor2 Factor3 Factor4
SS loadings 2.592 1.977 1.638 1.423
Proportion Var 0.236 0.180 0.149 0.129
Cumulative Var 0.236 0.415 0.564 0.694

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 24.26 on 17 degrees of freedom.
The p-value is 0.113
```

Hasil output diatas menunjukkan bahwa empat faktor yang dihasilkan dapat menjelaskan 69,4% total variansi data. Variabel-variabel utama penyusun faktor tersebut adalah

- Faktor 1 : variabel x9, x16, dan x18,
- Faktor 2 : variabel x7, x10, dan x12,
- Faktor 3 : variabel x8, dan x14,
- Faktor 4 : variabel x6 dan x13.

Output tersebut juga menunjukkan bahwa ada satu variabel yang sebaiknya dihilangkan dari analisis karena hasil rotasi masih masuk dalam dua faktor, yaitu x11 yang menyusun Faktor 1 dan 4.

R juga memberikan fasilitas untuk menampilkan hasil Analisis Faktor lebih mudah untuk diinterpretasi, yaitu dengan cara mengurutkan nilai loading faktor pada variabel penyusun faktor. Berikut ini adalah perintah **print** untuk menampilkan hasil Analisis Faktor dan output yang dihasilkan.

```
> print(FA, digits = 3, cutoff = 0.4, sort = TRUE)

Call:
factanal(x = ~x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13 + x14 + x16 + x18,
  factors = 4, data = hbat, scores = "regression", rotation = "varimax")

Uniquenesses:
      x6      x7      x8      x9      x10      x11      x12      x13      x14      x16      x18
0.682 0.360 0.228 0.178 0.679 0.005 0.017 0.636 0.163 0.347 0.076

Loadings:
      Factor1 Factor2 Factor3 Factor4
x9      0.884
x16     0.794
x18     0.928
x7              0.793
x10             0.521
x12             0.974
x8                0.872
x14               0.894
x6                  0.557
x11     0.502        0.856
x13                -0.514

      Factor1 Factor2 Factor3 Factor4
SS loadings    2.592    1.977    1.638    1.423
Proportion Var  0.236    0.180    0.149    0.129
Cumulative Var  0.236    0.415    0.564    0.694

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 24.26 on 17 degrees of freedom.
The p-value is 0.113
```

12.2. Analisis Diskriminan

Analisis Diskriminan merupakan salah satu metode dependensi dalam analisis multivariat yang biasanya digunakan untuk evaluasi klasifikasi objek. Sifat data yang digunakan dalam analisis ini adalah non-metrik pada variabel dependen (biasanya berupa kode group objek) dan metrik pada kelompok variabel independen. Tujuan dari analisis ini adalah mendapatkan suatu fungsi (disebut fungsi diskriminan) yang dapat digunakan untuk memisahkan objek sesuai dengan group atau klasifikasinya. Fungsi ini selanjutnya dapat juga digunakan untuk memprediksi group dari suatu objek baru yang diamati (Sharma, 1996).

R menyediakan fasilitas untuk Analisis Diskriminan Linear dan Kuadratik pada library **MASS** dengan perintah **lda** (linear) dan **qda** (kuadratik). Sebagai contoh, data **iris** yang sudah ada di paket **R** akan digunakan sebagai studi kasus untuk Analisis Diskriminan Linear. Berikut adalah contoh perintah dan output Analisis Diskriminan Linear pada data **iris** tersebut.

```
> Iris <- data.frame(rbind(iris3[,1], iris3[,2], iris3[,3]),
+                   Sp = rep(c("s","c","v"), rep(50,3)))
> train <- sample(1:150, 75) # Sampel 75 data dari 150
> table(Iris$Sp[train])      # 75 data sampel yang terpilih

  c  s  v
29 22 24

> z <- lda(Sp ~ ., Iris, prior = c(1,1,1)/3, subset = train)
> z                          # Menampilkan hasil Analisis Diskriminan

Call:
lda(Sp ~ ., data = Iris, prior = c(1, 1, 1)/3,
subset = train)

Prior probabilities of groups:
      c      s      v
0.3333333 0.3333333 0.3333333

Group means:
      Sepal.L. Sepal.W. Petal.L. Petal.W.
c  5.986207  2.765517  4.293103  1.327586
s  4.954545  3.400000  1.481818  0.250000
v  6.683333  2.958333  5.654167  2.016667

Coefficients of linear discriminants:
              LD1              LD2
Sepal.L.  0.8690984  0.07142847
Sepal.W.  1.5296128 -2.41239510
Petal.L. -1.7362163  0.89825836
Petal.W. -4.0571327 -2.94769154

Proportion of trace:
      LD1      LD2
0.9915 0.0085
```

Perintah Analisis Diskriminan Linear pada **script** diatas diaplikasikan pada 75 data sampel (**train**) dan sisanya digunakan untuk validasi apakah fungsi diskriminan yang diperoleh dapat memprediksi dengan tepat group dari 75 data sisanya. Taksiran nilai koefisien dari dua fungsi linear diskriminan yang dihasilkan dapat dilihat pada output tersebut.

```
> predict(z, Iris[-train, ])$class
[1] s s s s s s s s s s s s s s s s s s s s s s s s s s s s c c c c c c c c v c
[39] c c c c c c c c c c v v v v v v v v v v v v v v v c v c v v v v v v v v v
Levels: c s v
```

```
> table(Iris$Sp[train], predict(z, Iris[train, ])$class)
```

	c	s	v
c	28	0	1
s	0	22	0
v	0	0	24

```
> table(Iris$Sp[-train], predict(z, Iris[-train, ])$class)
```

	c	s	v
c	20	0	1
s	0	28	0
v	2	0	24

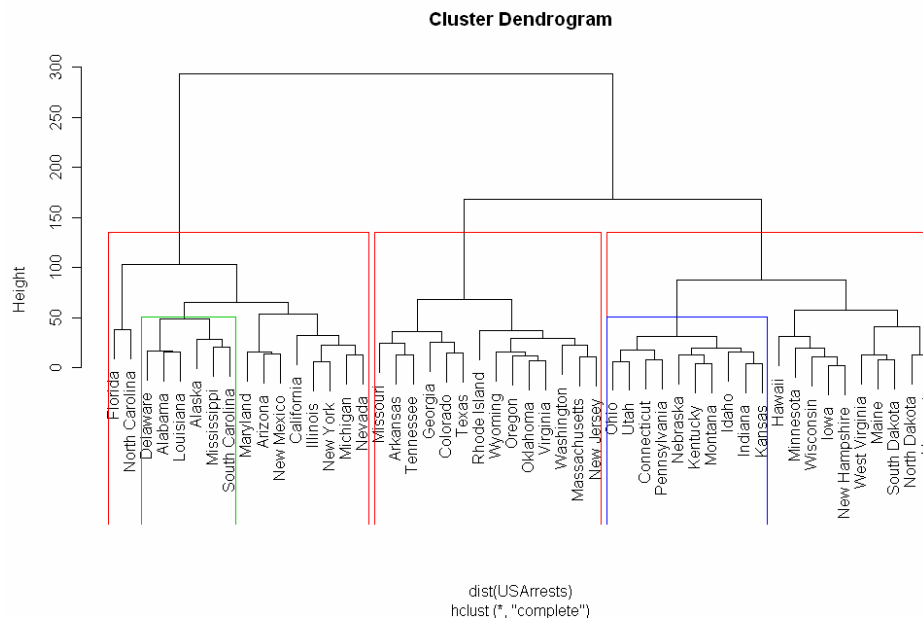
12.3. Analisis Cluster

Analisis Kluster K-mean...
Analisis Kluster Hierarkis...
Ringkas pengklasteran Hierarkis...
Tambahkan pengklasteran Hierarkis pada data...

Pada bagian ini, penjelasan tentang penggunaan **R** untuk Analisis Cluster akan difokuskan pada pemakaian perintah langsung di **R-Console** atau **command line**. Perintah **hclust** dapat digunakan untuk implementasi Analisis Cluster dengan beberapa pilihan metode *agglomeration*, yaitu "**ward**", "**single**", "**complete**", "**average**", "**mcquitty**", "**median**" atau "**centroid**". Misalkan akan diterapkan Analisis Cluster untuk mendapatkan kelompok negara-negara bagian di Amerika pada data **USArrest** yang sudah tersedia di paket **R**. Berikut ini adalah **script** dan hasil output dari Analisis Cluster Hirarki dengan menggunakan ukuran dissimilaritas dan metode **average**.

```
> hca <- hclust(dist(USArrests))
> plot(hca)
> rect.hclust(hca, k=3, border="red")
> x <- rect.hclust(hca, h=50, which=c(2,7), border=3:4)
```

Berikut ini adalah output dendrogram berdasarkan **script** diatas, yang dapat digunakan untuk menentukan jumlah kelompok yang akan dianalisis lanjut.



Gambar 12.1. Dendrogram untuk negara-negara bagian Amerika berdasarkan Analisis Cluster Hirarki metode "**average**"

Untuk mengetahui keanggotaan group atau kelompok yang dihasilkan dalam Analisis Cluster Hirarki diatas, **R** menyediakan perintah **cutree** untuk menampilkannya. Berikut ini adalah **script** dan hasil output tentang keanggotaan setiap obyek dengan Analisis Cluster Hirarki.

```
> hca <- hclust(dist(USArrests))
> cutree(hca, k=1:5)  # k = 1 adalah kelompok trivial
```

	1	2	3	4	5
Alabama	1	1	1	1	1
Alaska	1	1	1	1	1
Arizona	1	1	1	1	1
Arkansas	1	2	2	2	2
California	1	1	1	1	1
Colorado	1	2	2	2	2
Connecticut	1	2	3	3	3
Delaware	1	1	1	1	1
Florida	1	1	1	4	4
.....					
.....					
.....					
Utah	1	2	3	3	3
Vermont	1	2	3	3	5
Virginia	1	2	2	2	2
Washington	1	2	2	2	2
West Virginia	1	2	3	3	5
Wisconsin	1	2	3	3	5
Wyoming	1	2	2	2	2

```
> ## Perbandingan 2 dan 3 group hasil Cluster
> g24 <- cutree(hca, k = c(2,4))
> table(g24[, "2"], g24[, "4"])
```

	1	2	3	4
1	14	0	0	2
2	0	14	20	0

Output diatas menunjukkan hasil-hasil pengelompokan dengan menggunakan jumlah kelompok 1 sampai dengan 5. Hasil perbandingan untuk 2 kelompok dan 4 kelompok juga ditampilkan pada output diatas. Jika menggunakan 4 kelompok, maka 2 anggota yang semula di group 1 pada analisis dengan 2 kelompok menjadi group 4, sedangkan 20 anggota yang semula di group 2 pada analisis dengan 2 kelompok menjadi group 3.

BAB 13

REGRESI NONPARAMETRIK DAN ESTIMASI DENSITAS

Pada dekade terakhir ini, pemodelan statistika nonparametrik merupakan salah satu metode statistika yang berkembang dengan pesat seiring dengan perkembangan komputasi. Bab ini akan menjelaskan penggunaan **R** untuk pemodelan regresi dan estimasi densitas nonparametrik, khususnya penggunaan kernel dan spline.

13.1. Estimasi Densitas dengan Kernel

Estimasi nonparametrik dari fungsi densitas probabilitas merupakan suatu topik yang luas. Beberapa buku yang membahas tentang hal ini adalah Silverman (1986), Härdle (1991), Scott (1992), serta Wand dan Jones (1995). Metode estimasi yang dibahas dalam bagian ini adalah metode estimasi nonparametrik dengan kernel.

Perintah untuk membuat histogram yaitu **hist** dengan argumen **freq=FALSE** adalah juga suatu estimator dari fungsi densitas. Jika tiap-tiap titik tengah dari kelas di masing-masing histogram dihubungkan maka akan diperoleh estimator fungsi densitas dalam bentuk poligon frekuensi. Berikut ini adalah contoh **script** untuk mendapatkan histogram dalam frekuensi relatif (probabilitas) yang dapat digunakan sebagai estimator dari fungsi densitas pada suatu data, yaitu **geyser\$duration** yang sudah tersedia di **R**.

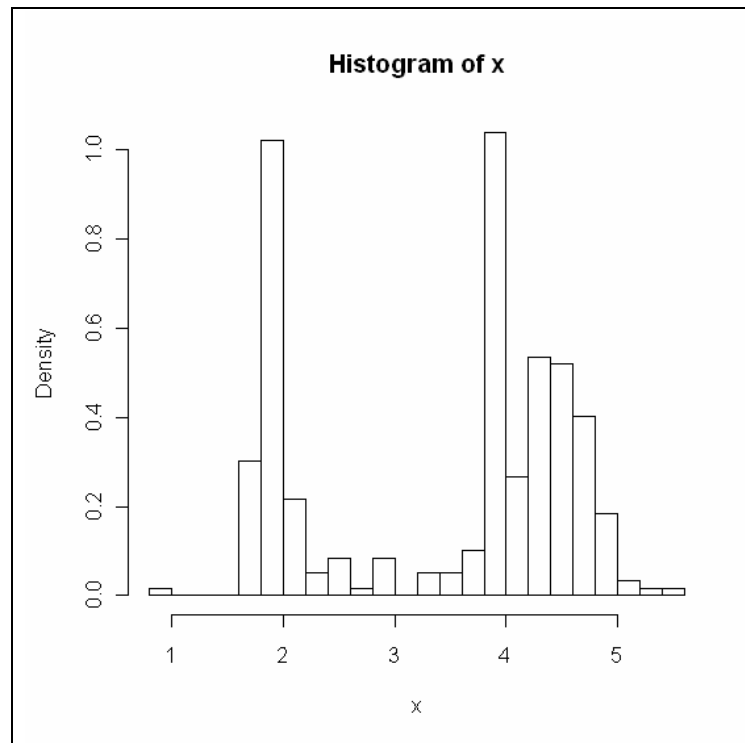
```
> data(geyser, package="MASS")
> x <- geyser$duration
> hist(x, breaks=22, freq=FALSE,
      main = "Smoothing with Gaussian Kernel")
```

Output dari **script** ini dapat dilihat pada Gambar 13.1. Dari gambar tersebut dapat dijelaskan bahwa data **geyser\$duration** bersifat bimodal, yaitu mempunyai dua macam puncak di sekitar angka 2 dan 4.

R menyediakan fasilitas estimasi nonparametrik dari fungsi densitas probabilitas dengan perintah **density** di *library stats* dan beberapa perintah di *library KernSmooth*. Secara matematis, implementasi penghalus densitas dengan kernel adalah

$$\hat{f}(x) = \frac{1}{b} \sum_{j=1}^n K\left(\frac{x - x_j}{b}\right)$$

untuk suatu sampel x_1, x_2, \dots, x_n , dengan $K(\cdot)$ suatu kernel tertentu dan b adalah *bandwith* yang digunakan.

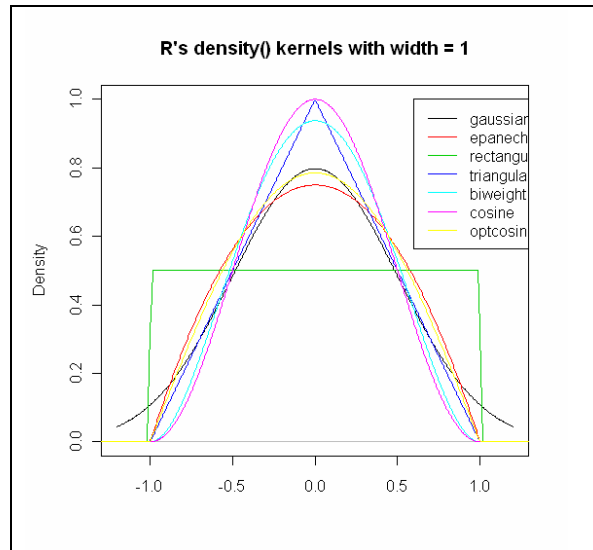


Gambar 13.1. Output histogram frekuensi relatif pada data **duration**

Pada bagian ini, penjelasan tentang estimasi nonparametrik dari suatu fungsi densitas akan ditekankan pada penggunaan perintah-perintah yang ada di *library KernSmooth*. Beberapa jenis kernel yang tersedia di *library KernSmooth* dapat dilihat pada tabel berikut ini.

Argumen Kernel	Keterangan Jenis Kernel
"normal"	Kernel <i>Gaussian</i> (pilihan <i>default</i>)
"box"	Kernel <i>Rectangular box</i>
"epanech"	Kernel <i>Epanechnikov</i> (<i>the centred beta(2,2) density</i>)
"biweight"	Kernel <i>Biweight</i> (<i>the centred beta(3,3) density</i>)
"triweight"	Kernel <i>Triweight</i> (<i>the centred beta(4,4) density</i>)

Berikut ini adalah contoh grafik dari bentuk-bentuk kernel untuk estimasi nonparametrik dari suatu fungsi densitas.



Gambar 13.2. Grafik dari berbagai kernel untuk estimasi densitas

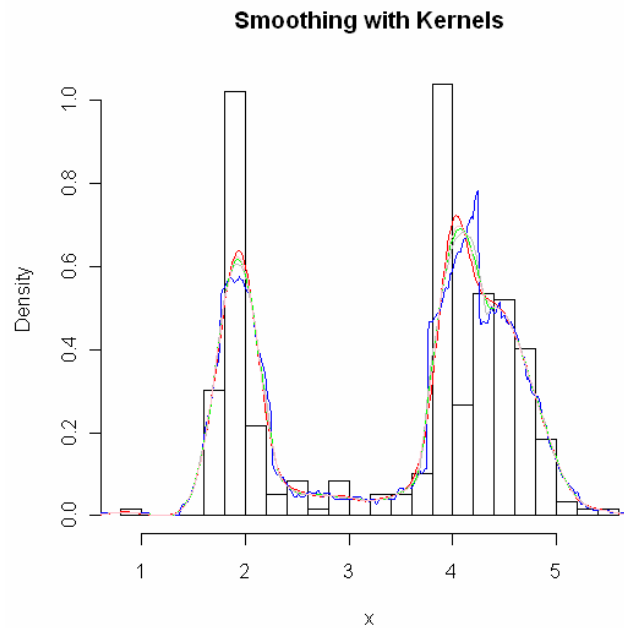
Perintah **dpik** pada *library KernSmooth* dapat digunakan untuk mendapatkan nilai *bandwith* yang optimal pada suatu kernel yang diimplementasikan pada suatu data. Berikut ini adalah **script** untuk implementasi estimasi nonparametrik dari fungsi densitas pada data **geyser\$duration** yang sudah tersedia di **R**, dengan menggunakan pilihan *bandwith* optimal.

```
> data(geyser, package="MASS")
> x <- geyser$duration
> h.n <- dpik(x, kernel="normal") # Pemilihan bandwith yang optimal
> est.n <- bkde(x, kernel="normal", bandwidth=h.n)
> h.b <- dpik(x, kernel="box")
> est.b <- bkde(x, kernel="box", bandwidth=h.b)
> h.e <- dpik(x, kernel="epanech")
> est.e <- bkde(x, kernel="epanech", bandwidth=h.e)
> h.bi <- dpik(x, kernel="biweight")
> est.bi <- bkde(x, kernel="biweight", bandwidth=h.bi)
> h.tri <- dpik(x, kernel="triweight")
> est.tri <- bkde(x, kernel="triweight", bandwidth=h.tri)
```

Tampilan besarnya *bandwith* optimal dan hasil perbandingan grafik dari estimator-estimator densitas pada data **geyser\$duration** dapat dilakukan dengan menggunakan **script** berikut ini.

```
> # Perbandingan bandwidth optimal pada masing-masing kernel
> hopt <- cbind(h.n,h.b,h.e,h.bi,h.tri)
> hopt
      h.n      h.b      h.e      h.bi      h.tri
[1,] 0.1438196 0.2502543 0.3183884 0.3771834 0.4283099
> win.graph()
> hist(x, breaks=22, freq=FALSE,main = "Smoothing with Kernels")
> lines(est.n, col='red')      # densitas dengan kernel normal
> lines(est.b, col='blue')     # densitas dengan kernel rectangular
> lines(est.e, col='gray')     # densitas dengan kernel epanechnikov
> lines(est.bi, col='green')   # densitas dengan kernel biweight
> lines(est.tri, col='pink')   # densitas dengan kernel triangular
```

Output dari **script** diatas adalah estimasi dari bentuk fungsi densitas probabilitas seperti pada gambar di bawah ini.

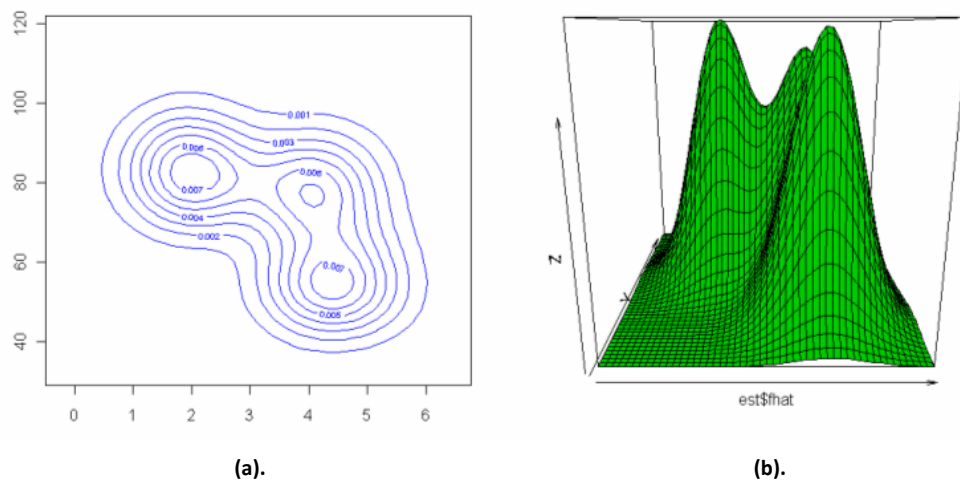


Gambar 13.3. Grafik dari berbagai kernel untuk estimasi densitas

Sebagai tambahan, pada *library KernSmooth* juga tersedia fasilitas untuk analisis kernel dua dimensi, yaitu dengan perintah **bkde2D**. Berikut ini adalah contoh **script** tentang analisis kernel dua dimensi pada data **geyser** yang sudah ada di **R**.

```
> data(geyser, package="MASS")
> x <- cbind(geyser$duration, geyser$waiting)
> est <- bkde2D(x, bandwidth=c(0.7,7))
> contour(est$x1, est$x2, est$fhat)
> persp(est$fhat)
```

Berikut ini adalah output dari **script** diatas.



Gambar 13.4. Grafik dari estimasi densitas dua dimensi pada data **geyser**

13.2. Regresi Nonparametrik dengan Kernel

Dalam praktek, seringkali dijumpai permasalahan keterkaitan antara variabel independen dan dependen yang bentuk keterkaitannya tidak diketahui secara pasti atau hanya ada sedikit informasi tentang bentuk keterkaitan tersebut. Regresi nonparametrik adalah suatu metode statistika yang banyak digunakan untuk menganalisis hubungan antara variabel independen dan dependen yang bentuk hubungan antar variabel tersebut tidak diketahui. Ada beberapa metode dalam regresi nonparametrik, antara lain dengan penghalusan (*smoothing*) kernel dan spline.

R menyediakan fasilitas estimasi regresi nonparametrik dengan kernel pada beberapa perintah di *library KernSmooth*. Secara matematis, implementasi regresi nonparametrik dengan kernel adalah

$$\hat{y}_i = \frac{\sum_{j=1}^n y_j K\left(\frac{x_i - x_j}{b}\right)}{\sum_{j=1}^n K\left(\frac{x_i - x_j}{b}\right)}$$

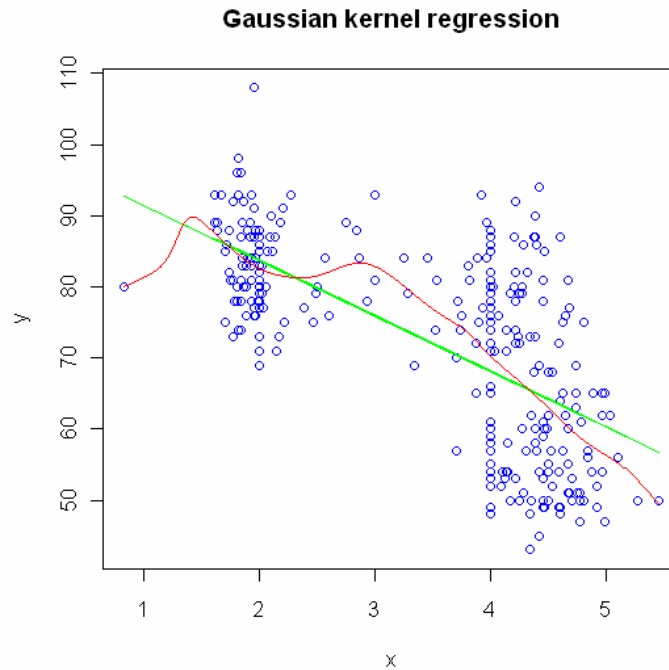
dengan b adalah parameter *bandwidth*, dan $K()$ suatu fungsi kernel seperti pada estimasi densitas sebelumnya. Ada dua perintah utama di *library KernSmooth* yang dapat digunakan untuk estimasi regresi nonparametrik dengan kernel, yaitu :

Perintah	Keterangan
dpill	menggunakan metodologi <i>direct plug-in</i> untuk memilih <i>bandwidth</i> pada suatu estimasi regresi kernel <i>Gaussian</i> linear lokal, seperti yang dideskripsikan oleh Ruppert, Sheather dan Wand (1995).
locpoly	estimasi suatu fungsi densitas probabilitas, fungsi regresi atau turunannya dengan menggunakan polinomial lokal.

Berikut ini adalah **script** untuk implementasi estimasi regresi nonparametrik dengan kernel *Gaussian* pada data **geyser\$duration** sebagai **x** dan **geyser\$waiting** sebagai **y** yang sudah tersedia di **R**, dengan menggunakan pilihan *bandwidth* optimal.

```
> data(geyser, package = "MASS")
> x <- geyser$duration
> y <- geyser$waiting
> win.graph()
> plot(x, y, col="blue")
> h.opt <- dpill(x, y) # Selection the optimal bandwidth
> fit.opt <- locpoly(x, y, bandwidth = h.opt)
> lines(fit.opt, col="red")
> title(main="Gaussian kernel regression with optimal bandwidth")
```

Output dari **script** diatas adalah garis halus (*smooth*) dari estimasi regresi nonparametrik dengan kernel *Gaussian* seperti berikut (garis warna merah).



Gambar 13.5. Grafik dari estimasi regresi nonparametrik dengan kernel *Gaussian*

13.3. Regresi Nonparametrik dengan Spline

Seperti pada estimasi regresi nonparametrik dengan kernel, **R** juga menyediakan fasilitas estimasi regresi nonparametrik dengan spline, yaitu dengan menggunakan perintah **smooth.spline**. Misalkan diketahui ada n pasangan data (x_i, y_i) . Suatu penghalusan (*smoothing*) spline meminimumkan suatu kompromi antara *fit* (taksiran) dan derajat dari penghalus (*smoothness*) dalam bentuk

$$\sum w_i [y_i - f(x_i)]^2 + \lambda \int (f''(x))^2 dx$$

pada semua fungsi (terukur yang dapat diturunkan dua kali) f . Ini adalah suatu spline kubik dengan knot-knot pada x_i , tetapi tidak menginterpolasi titik-titik data untuk $\lambda > 0$ dan derajat dari fit dikontrol oleh λ . Penentuan λ dapat ditetapkan tertentu atau dipilih secara otomatis dengan metode *cross-validation*.

Perintah dan argumen untuk aplikasi regresi nonparametrik dengan spline kubik pada **R** adalah sebagai berikut.

```
smooth.spline(x, y = NULL, w = NULL, df, spar = NULL,
              cv = FALSE, all.knots = FALSE, nknots = NULL,
              keep.data = TRUE, df.offset = 0, penalty = 1,
              control.spar = list())
```

Berikut ini adalah penjelasan tentang pilihan argumen yang dapat digunakan dalam perintah **smooth.spline**.

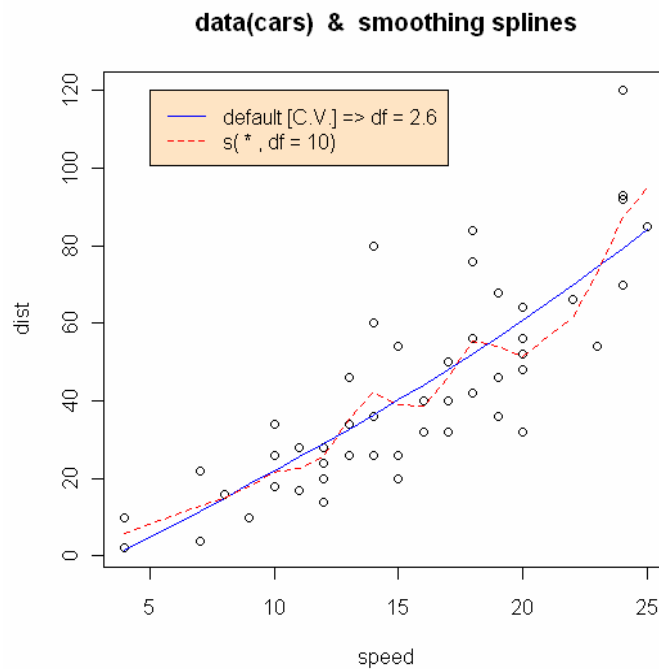
Argumen	Keterangan
x	a vector giving the values of the predictor variable, or a list or a two-column matrix specifying x and y.
y	responses. If y is missing, the responses are assumed to be specified by x.
w	optional vector of weights of the same length as x; defaults to all 1.
df	the desired equivalent number of degrees of freedom (trace of the smoother matrix).
spar	smoothing parameter, typically (but not necessarily) in $(0,1]$. The coefficient λ of the integral of the squared second derivative in the fit (penalized log likelihood) criterion is a monotone function of spar, see the details below.
cv	ordinary (TRUE) or 'generalized' cross-validation (GCV) when FALSE.
all.knots	if TRUE, all distinct points in x are used as knots. If FALSE (default), a subset of $x[j]$ is used, specifically $x[j]$ where the nknots indices are evenly spaced in $1:n$, see also the next argument nknots.
nknots	integer giving the number of knots to use when all.knots=FALSE. Per default, this is less than n , the number of unique x values for $n > 49$.
keep.data	logical specifying if the input data should be kept in the result. If TRUE (as per default), fitted values and residuals are available from the result.
df.offset	allows the degrees of freedom to be increased by df.offset in the GCV criterion.
penalty	the coefficient of the penalty for degrees of freedom in the GCV criterion.
control.spar	optional list with named components controlling the root finding when the smoothing parameter spar is computed, i.e., missing or NULL, see below. Note that this is partly <i>experimental</i> and may change with general spar computation improvements!

Di bawah ini adalah **script** untuk implementasi regresi nonparametrik dengan spline kubik pada data **cars** yaitu variabel **speed** sebagai **x** dan **dist** sebagai **y** yang sudah tersedia di **R**, serta output yang dihasilkan.

```
> attach(cars)
> plot(speed, dist, main = "data(cars) & smoothing splines")
> cars.spl <- smooth.spline(speed, dist)
> (cars.spl)
Call:
smooth.spline(x = speed, y = dist)

Smoothing Parameter spar= 0.7801305 lambda= 0.1112206 (11 iterations)
Equivalent Degrees of Freedom (Df): 2.635278
Penalized Criterion: 4187.776
GCV: 244.1044

> lines(cars.spl, col = "blue")
> lines(smooth.spline(speed, dist, df=10), lty=2, col = "red")
> legend(5,120,c(paste("default [C.V.] => df =",round(cars.spl$df,1)),
+               "s( *, df = 10)"), col = c("blue","red"), lty = 1:2, bg='bisque')
```

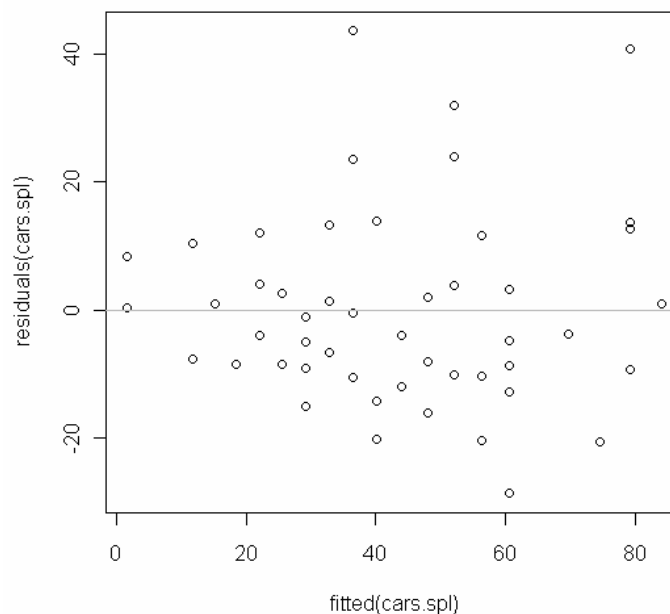


Gambar 13.6. Grafik hasil regresi spline kubik pada data **cars**

Pengecekan kebaikan model regresi nonparametrik spline kubik ini dapat dilakukan dengan melihat analisis residual model. Berikut ini adalah **script** untuk plot residual dari model di atas.

```
> ## Residual (Tukey Anscombe) plot:  
> plot(residuals(cars.spl) ~ fitted(cars.spl))  
> abline(h = 0, col="gray")
```

Output yang dihasilkan dari **script** ini adalah seperti gambar berikut ini.



Gambar 13.7. Grafik analisis residual dari regresi nonparametrik spline kubik

Hasil di atas menunjukkan bahwa residual cenderung mempunyai pola yang tidak homogen, yaitu cenderung membesar seiring meningkatnya nilai prediksi (*fitted*).

Sebagai tambahan, berikut ini adalah contoh **script** lain untuk implementasi regresi nonparametrik dengan **spline kubik** pada suatu data simulasi, dengan berbagai parameter smoothing (**spar**) dan prediksi pada data *training* dan *testing*.

```

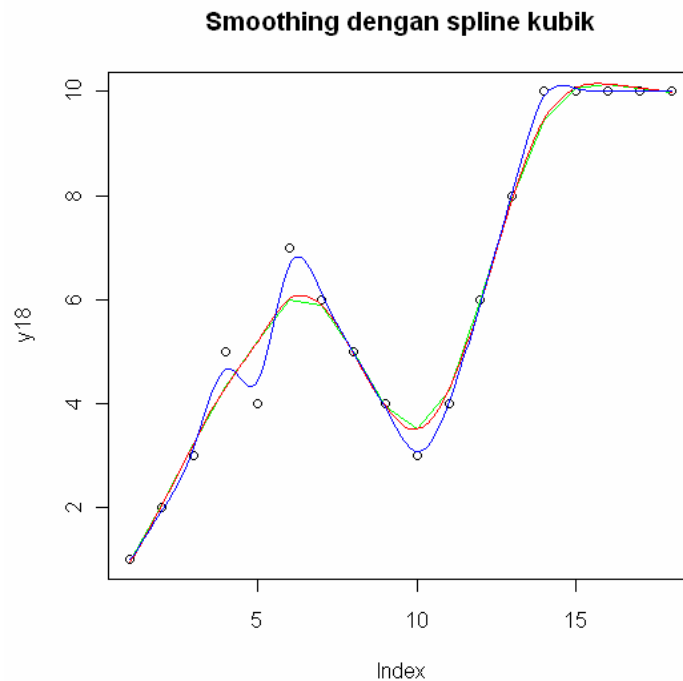
# Contoh smoothing pada data simulasi simulasi
y18 <- c(1:3,5,4,7:3,2*(2:5),rep(10,4))
xx <- seq(1,length(y18), len=201)

# Regresi spline kubik dengan x=1:18 dan y=y18
(s2 <- smooth.spline(y18))          # Smoothing parameter dgn GCV
(s02 <- smooth.spline(y18, spar = 0.2)) # Smoothing parameter 0.2

# Plot perbandingan hasil regresi spline kubik dengan berbagai x dan spar
plot(y18, main="Smoothing dengan spline kubik")
lines(s2, col = "green")             # Hasil prediksi dengan x
lines(predict(s2, xx), col = "red")  # Hasil prediksi "GCV" dengan xx
lines(predict(s02, xx), col = "blue") # Hasil prediksi "spar=0.2" dengan x

```

Output perbandingan grafik dari **script** ini dapat dilihat pada Gambar 13.8. Perhatikan pilihan warna pada **script** untuk mengetahui pilihan regresi kubik spline yang digunakan.



Gambar 13.8. Output regresi spline kubik pada data simulasi

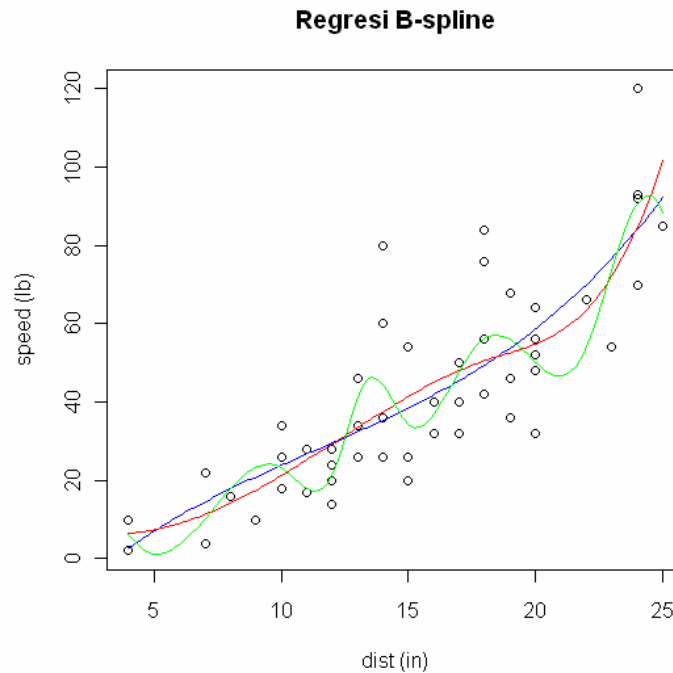
Selain **spline kubik**, paket **R** juga menyediakan fasilitas untuk implementasi jenis **spline** yang lain. Ada banyak *library* yang dapat digunakan, salah satunya adalah *library splines* yang menyediakan fasilitas untuk **B-spline**. Berikut ini adalah contoh **script** untuk implementasi regresi nonparametrik dengan **B-spline** pada data **cars**, dengan variabel **speed** sebagai **x** dan **dist** sebagai **y** yang sudah tersedia di **R**.

```
library(splines)
bs(cars$speed, df = 3)
summary(fm3 <- lm(dist ~ bs(speed, df = 3), data = cars))
bs(cars$speed, df = 5)
summary(fm5 <- lm(dist ~ bs(speed, df = 5), data = cars))
bs(cars$speed, df = 10)
summary(fm10 <- lm(dist ~ bs(speed, df = 10), data = cars))
## Contoh menyimpan dan menampilkan prediksi
plot(cars, xlab = "dist (in)", ylab = "speed (lb)", main="Regresi B-spline")
ht <- seq(4, 25, length.out = 200)
lines(ht, predict(fm3, data.frame(speed=ht)), col="blue")
lines(ht, predict(fm5, data.frame(speed=ht)), col="red")
lines(ht, predict(fm10, data.frame(speed=ht)), col="green")
```

Ada dua output utama yang akan dihasilkan dari **script** tersebut, yaitu hasil-hasil estimasi **B-spline** yang terdiri dari nilai prediksi, **knots**, dan koefisien dari model regresi spline, serta output grafik hasil prediksi. Berikut adalah sebagian hasil dari **script** diatas.

```
> bs(cars$speed, df = 5)
      1      2      3      4      5
[1,] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
.....
[50,] 0.000000000 0.000000000 0.000000000 0.000000000 1.000000000
attr(,"degree")
[1] 3
attr(,"knots")
33.33333% 66.66667%
  13  18
attr(,"Boundary.knots")
[1] 4 25
attr(,"intercept")
[1] FALSE
attr(,"class")
[1] "bs" "basis"
```


Hasil output grafik hasil prediksi regresi nonparametrik dengan **B-spline** untuk berbagai derajat spline dapat dilihat pada gambar berikut ini. Untuk mengetahui perbedaan efek dari derajat spline terhadap hasil *smoothing* yang diperoleh, perhatikan pilihan warna pada **script** diatas.



Gambar 13.9. Grafik hasil regresi **B-spline** pada data **cars**

13.4. Jenis-jenis Basis Spline

Pada bagian sebelumnya telah diilustrasikan hasil regresi nonparametrik dengan spline **kubik** dan **B-spline**. Secara umum ada beberapa jenis spline yang dapat digunakan dalam regresi nonparametrik. **R** menyediakan fasilitas untuk mengetahui macam-macam spline, dikenal **fungsi basis**, yang dapat digunakan untuk regresi nonparametrik, yaitu pada *library* **fda**. Dalam *library* ini disediakan fasilitas untuk membuat delapan macam fungsi basis spline, yaitu basis **B-spline**, **Constant**, **Exponential**, **Fourier**, **Monomial**, **Polygonal**, **Polynomial**, dan **Power Basis Object**.

Fungsi-fungsi yang ada di *library* **fda** dikembangkan untuk mendukung analisis data fungsional seperti yang digambarkan oleh Ramsay dan Silverman (2005). Berikut ini adalah tabel yang berisi perintah-perintah untuk membuat fungsi-fungsi basis spline yang dapat diaplikasikan untuk pemodelan regresi nonparametrik.

Perintah	Keterangan
create.bspline.basis	Untuk membuat suatu <i>B-spline Basis</i>
create.constant.basis	Untuk membuat suatu <i>Constant Basis</i>
create.exponential.basis	Untuk membuat suatu <i>Exponential Basis</i>
create.fourier.basis	Untuk membuat suatu <i>Fourier Basis</i>
create.monomial.basis	Untuk membuat suatu <i>Monomial Basis</i>
create.polygonal.basis	Untuk membuat suatu <i>Polygonal Basis</i>
create.polynomial.basis	Untuk membuat suatu <i>Polynomial Basis</i>
create.power.basis	Untuk membuat suatu <i>Power Basis Object</i>

Berikut ini adalah contoh **script** untuk membuat macam-macam basis **B-spline** yang dapat digunakan pada regresi nonparametrik.

```
# The simplest basis currently available with this function:
str(bspl1.1 <- create.bspline.basis(norder=1, breaks=0:1))
# 1 basis function, order 1 = degree 0 = step function:
# constant 1 between 0 and 1.

str(bspl1.2 <- create.bspline.basis(norder=1, breaks=c(0,.5, 1)))
# 2 bases, order 1 = degree 0 = step functions:
# (1) constant 1 between 0 and 0.5 and 0 otherwise
# (2) constant 1 between 0.5 and 1 and 0 otherwise.

str(bspl2.3 <- create.bspline.basis(norder=2, breaks=c(0,.5, 1)))
# 3 bases: order 2 = degree 1 = linear
# (1) line from (0,1) down to (0.5, 0), 0 after
# (2) line from (0,0) up to (0.5, 1), then down to (1,0)
# (3) 0 to (0.5, 0) then up to (1,1).

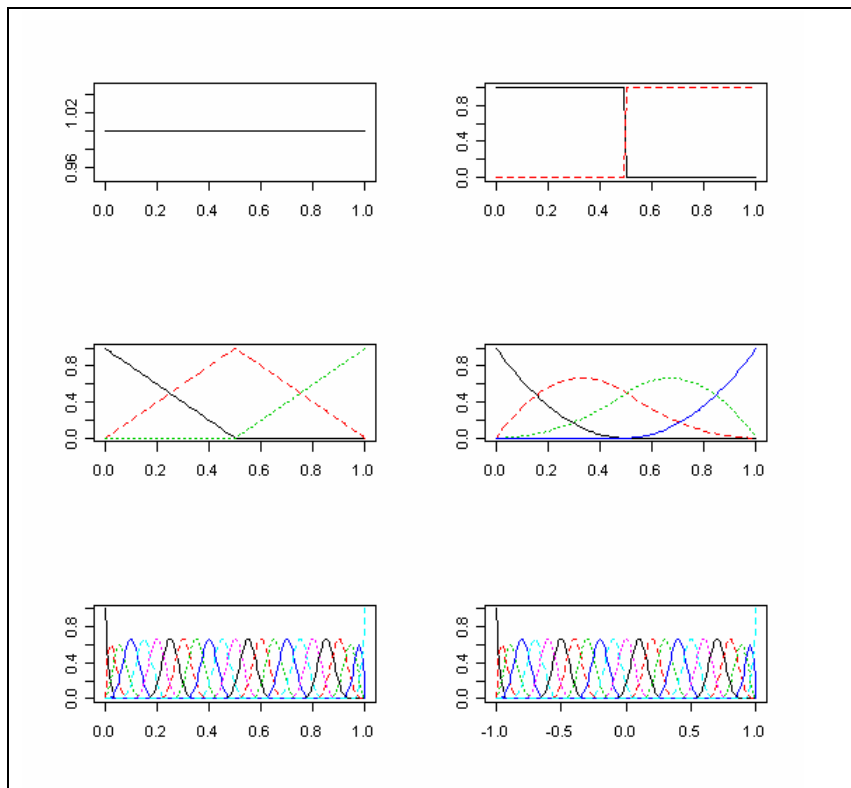
str(bspl3.4 <- create.bspline.basis(norder=3, breaks=c(0,.5, 1)))
# 4 bases: order 3 = degree 2 = parabolas.
# (1) (x-.5)^2 from 0 to .5, 0 after
# (2) 2*(x-1)^2 from .5 to 1, and a parabola
# from (0,0 to (.5, .5) to match
# (3 & 4) = complements to (2 & 1).
```

```
# Default b-spline basis
str(bSpl4.23 <- create.bspline.basis())
# Cubic bspline (norder=4) with nbasis=23,
# so nbreaks = nbasis-norder+2 = 21,
# 2 of which are rangeval, leaving 19 Interior knots.

str(bSpl4. <- create.bspline.basis(c(-1,1)))
# Same as bSpl4.23 but over (-1,1) rather than (0,1).

win.graph()
par(mfrow=c(3,2))
plot(bspl1.1); plot(bspl1.2); plot(bspl2.3)
plot(bspl3.4); plot(bSpl4.23); plot(bSpl4.)
```

Output dari **script** diatas adalah sebagai berikut.

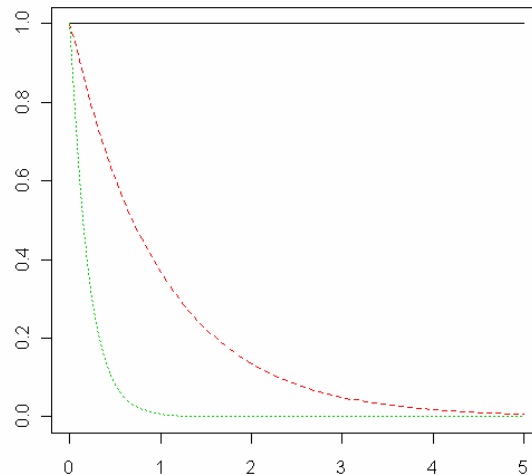


Gambar 13.10. Macam-macam fungsi basis B-spline

Basis *constant* adalah suatu basis spline yang menghasilkan suatu konstanta seperti yang terlihat pada plot pertama di Gambar 13.10 atau yang paling atas sebelah kiri. Fungsi basis yang lain adalah **exponential**, dan berikut adalah **script** untuk membuat basis spline **exponential**.

```
# Create an exponential basis over interval [0,5]
# with basis functions 1, exp(-t) and exp(-5t)
basisobj <- create.exponential.basis(c(0,5),3,c(0,-1,-5))
# plot the basis
plot(basisobj)
```

Hasil output grafik dari **script** ini adalah sebagai berikut.



Gambar 13.11. Macam-macam fungsi basis **exponential**

Selanjutnya akan dijelaskan tentang fungsi basis Fourier. Fungsi basis Fourier adalah suatu sistem yang biasanya digunakan pada fungsi-fungsi yang periodik. Fungsi basis Fourier yang pertama adalah suatu fungsi konstanta. Sedangkan yang lainnya adalah pasangan dari **sin** dan **cos** dengan pengali suatu *integer* dari periode dasar. Jumlah atau banyaknya fungsi basis yang dibangkitkan adalah selalu ganjil. Berikut adalah **script** untuk membuat fungsi basis Fourier.

```

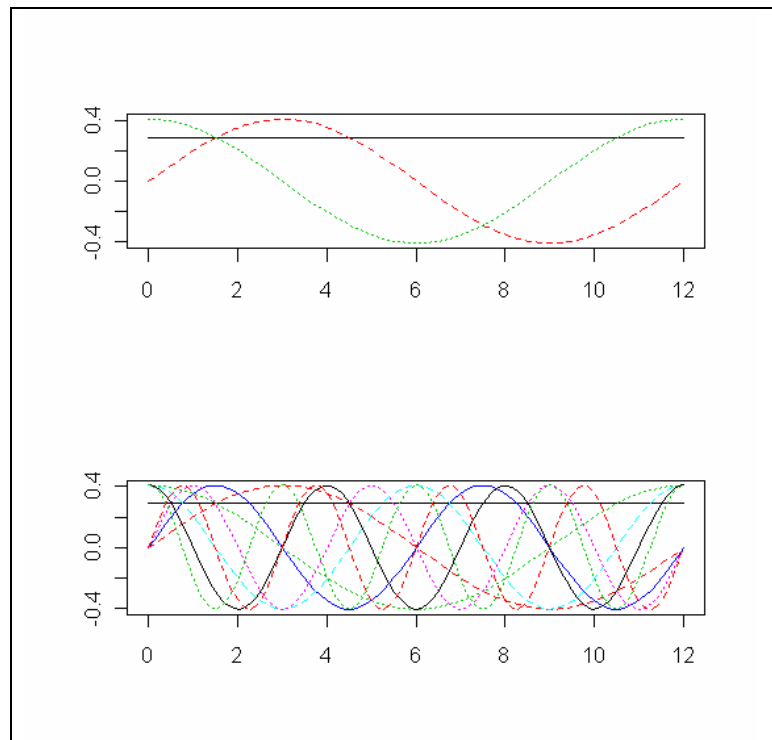
# Create a minimal Fourier basis for the monthly temperature data,
# using 3 basis functions with period 12 months.
monthbasis3 <- create.fourier.basis(c(0,12) )

# set up the Fourier basis for the monthly temperature data,
# using 9 basis functions with period 12 months.
monthbasis <- create.fourier.basis(c(0,12), 9, 12.0)

# plot the basis
win.graph()
par(mfrow=c(2,1))
plot(monthbasis3)
plot(monthbasis)

```

Output grafik dari **script** ini adalah sebagai berikut.



Gambar 13.12. Macam-macam fungsi basis **Fourier**

13.5. Rangkuman library untuk Aplikasi Kernel dan Spline

Pada bagian ini akan diberikan suatu rangkuman tentang beberapa *library* beserta keterangan tentang perintah-perintah dalam *library* tersebut yang disediakan **R** untuk aplikasi metode **kernel** dan **spline**.

- Tabel *library* untuk aplikasi **kernel**

library	Keterangan
feature	<i>Feature significance for multivariate kernel density estimation.</i>
GenKern	<i>Functions for generating and manipulating kernel density estimates.</i>
KernSmooth	Fungsi-fungsi untuk penghalusan (<i>smoothing</i>) <i>kernel</i> dan <i>density estimation</i> , seperti pada buku Wand dan Jones (1995) dengan judul "Kernel Smoothing".
kernlab	<i>Kernel-based machine learning methods for classification, regression, clustering, novelty detection, quantile regression and dimensionality reduction. Among other methods kernlab includes Support Vector Machines, Spectral Clustering, Kernel PCA and a QP solver.</i>
kerfdr	<i>Semi-parametric kernel-based approach to local fdr estimations</i>
ks	<i>Kernel density estimators and kernel discriminant analysis for multivariate data.</i>
locpol	<i>Kernel local polinomial regression.</i>
lokern	<i>Kernel regression smoothing with adaptive local or global plug-in bandwidth selection.</i>
MKLE	<i>Maximum kernel likelihood estimation.</i>
monreg	<i>Nonparametric monotone regression.</i>
monoProc	<i>Strictly monotone smoothing procedure, given fit in one or two variables.</i>
np	<i>Nonparametric kernel smoothing methods for mixed datatypes.</i>
sm	<i>Smoothing methods for nonparametric regression and density estimation.</i>

▪ Tabel *library* untuk aplikasi **spline**

library	Keterangan
assist	<i>A Suite of S-Plus Functions Implementing Smoothing Splines</i>
cobs99	<i>Constrained B-splines</i>
DierckxSpline	<i>R companion to "Curve and Surface Fitting with Splines"</i>
earth	<i>Build regression models using the techniques in Friedman's papers "Fast MARS" and "Multivariate Adaptive Regression Splines".</i>
fda	<i>These functions were developed to support functional data analysis as described in Ramsay, J. O. and Silverman, B. W. (2005) Functional Data Analysis. New York: Springer.</i>
gss	<i>A comprehensive package for structural multivariate function estimation using smoothing splines.</i>
kzs	<i>A collection of functions utilizing splines to construct a smooth estimate of a signal buried in noise.</i>
lmeSplines	<i>Add smoothing spline modelling capability to nlme. Fit smoothing spline terms in Gaussian linear and nonlinear mixed-effects models.</i>
logspline	<i>Routines for the logspline density estimation.</i>
MBA	<i>Scattered data interpolation with Multilevel B-Splines</i>
mda	<i>Mixture and flexible discriminant analysis, multivariate additive regression splines (MARS), BRUTO, ...</i>
polspline	<i>Routines for the polynomial spline fitting routines hazard regression, hazard estimation with flexible tails, logspline, lspec, polyclass, and polymars</i>
pspline	<i>Penalized Smoothing Splines. Smoothing splines with penalties on order m derivatives.</i>
sspline	<i>R package for Computing the Spherical Smoothing Splines</i>

BAB 14

MODEL NON-LINEAR

Pemodelan yang digunakan untuk menjelaskan hubungan nonlinear antar variabel dan beberapa prosedur pengujian untuk mendeteksi adanya keterkaitan nonlinear telah mengalami perkembangan yang sangat pesat pada beberapa dekade terakhir ini. Sebagai *overview* hal ini dapat dilihat antara lain pada buku Granger dan Terasvirta (1993). Perkembangan yang pesat ini juga terjadi dalam bidang pemodelan statistik secara umum. Pada bab ini akan dijelaskan tentang penggunaan paket **R** untuk pemodelan non-linear.

14.1. Estimasi Model Regresi Non-linear

Teori tentang model regresi non-linear secara lengkap dapat dilihat di Seber dan Wild (1989). Bentuk umum dari suatu model regresi non-linear adalah

$$y = \eta(\mathbf{x}, \boldsymbol{\beta}) + \varepsilon$$

dengan \mathbf{x} adalah suatu vektor kovariat, $\boldsymbol{\beta}$ adalah suatu vektor p -komponen parameter yang tidak diketahui, dan ε adalah suatu error yang $N(0, \sigma^2)$. Misalkan suatu model regresi non-linear dalam bentuk

$$y = \frac{\beta_1}{1 + \exp(\beta_2 + \beta_3 t)} + \varepsilon$$

akan diaplikasikan pada data **US.pop** yaitu tentang populasi di Amerika Serikat, dengan y adalah jumlah populasi dan t menyatakan tahun. Berikut ini adalah **script** untuk memanggil dan menampilkan data **US.pop** dalam suatu diagram pencar.

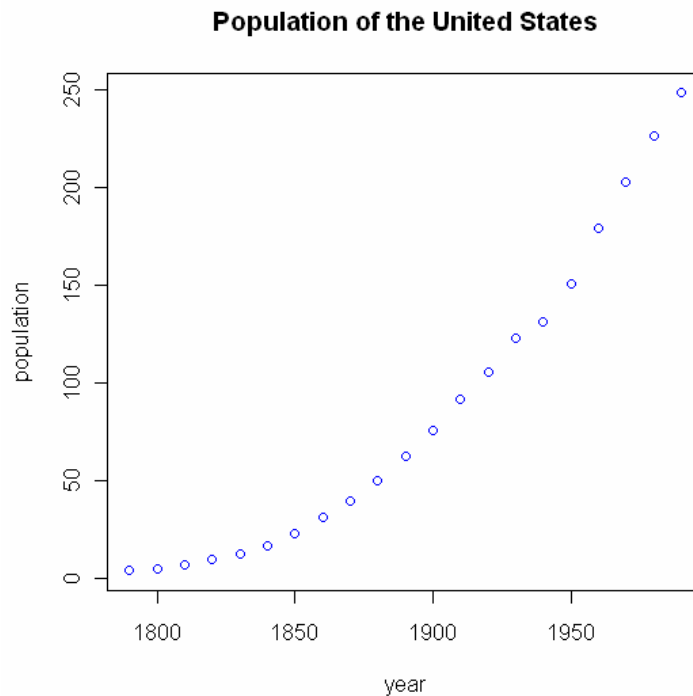
```
> library(car)
> data(US.pop)
> attach(US.pop)

The following object(s) are masked from US.pop ( position 3 ) :
population year

> US.pop
  year population
1  1790     3.929
2  1800     5.308
...   .....
21 1990    248.710

> plot(year, population, main='Population of the United States')
```


Plot antara t yang menyatakan tahun dan y tentang jumlah populasi di US dapat dilihat pada Gambar 14.1. Dari gambar ini dapat dijelaskan bahwa ada hubungan yang non-linear antara t dan y .



Gambar 14.1. Diagram pencar antara t dan y pada data **US.pop**

Dalam contoh kasus **US.pop** ini, vektor parameternya adalah $\beta = (\beta_1, \beta_2, \beta_3)^T$. Berikut ini adalah **script** untuk mendapatkan taksiran parameter pada model regresi non-linear diatas.

```
> time <- 0:20
> pop.mod <- nls(population ~ beta1/(1 + exp(beta2 + beta3*time)),
+   start=list(beta1=350, beta2=4.5, beta3=-0.3), trace=T)
13007.48 : 350.0 4.5 -0.3
609.5727 : 351.8074862 3.8405002 -0.2270578
365.4396 : 383.7045367 3.9911148 -0.2276690
.....
356.4001 : 389.1655126 3.9903457 -0.2266199
```

```
> summary(pop.mod)

Formula: population ~ beta1/(1 + exp(beta2 + beta3 * time))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
beta1 389.16551  30.81197   12.63 2.20e-10 ***
beta2  3.99035   0.07032   56.74 < 2e-16 ***
beta3 -0.22662   0.01086  -20.87 4.60e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.45 on 18 degrees of freedom

Number of iterations to convergence: 6
Achieved convergence tolerance: 1.492e-06

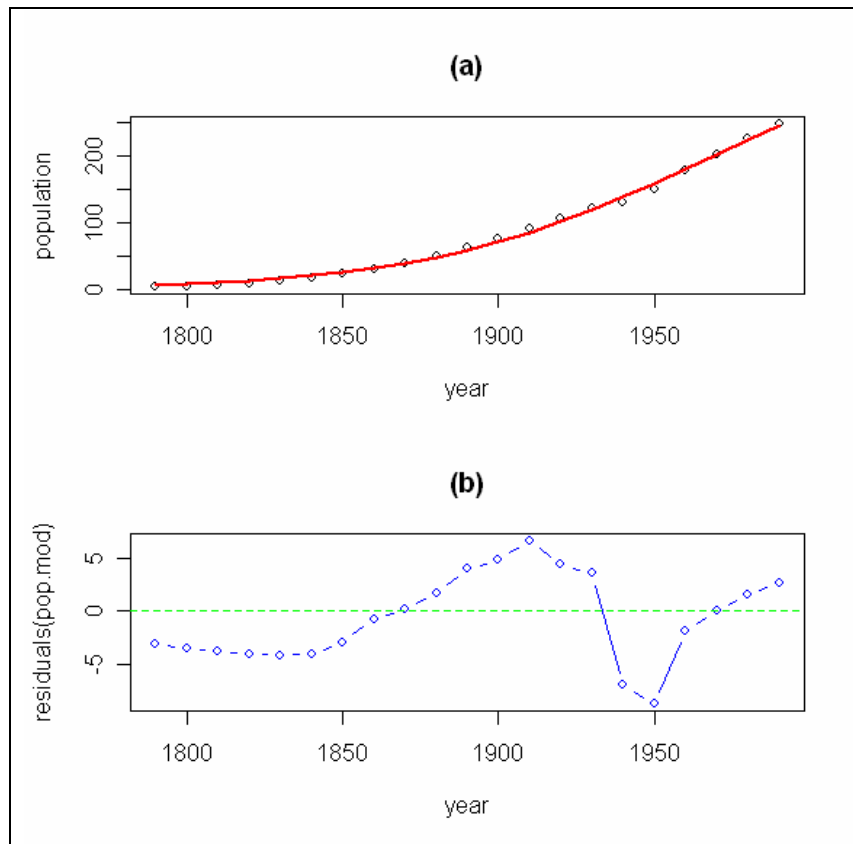
> par(mfrow=c(1,2))
> plot(year, population, main='(a)')
> lines(year, fitted.values(pop.mod), lwd=2, col=2)
> plot(year, residuals(pop.mod), type='b', main='(b)', col="blue")
> abline(h=0, lty=2, col="green")
```

Hasil diatas menunjukkan nilai-nilai taksiran dari parameter model regresi non-linear, sehingga secara lengkap model non-linear yang diperoleh dapat ditulis dalam bentuk

$$y = \frac{389.16551}{1 + \exp(3.99035 - 0.22662t)} + \varepsilon.$$

Output tersebut juga menunjukkan bahwa nilai taksiran ketiga parameter itu adalah signifikan secara statistik pada *alpha* 0.001. Sebagai catatan, *t* dalam model regresi non-linear ini menyatakan suatu kode dari tahun, yaitu 0 untuk tahun 1790, 1 untuk 1800, dan seterusnya. Pada output **summary(pop.mod)** juga ditampilkan nilai taksiran standar error dari residual, yaitu 4.45.

Pada bagian akhir dari **script** diatas berisi perintah untuk membuat plot antara nilai aktual dan prediksi secara bersama-sama, serta perintah untuk menyajikan plot residual model. Hasil lengkap plot perbandingan ini dapat dilihat pada Gambar 14.2a. Dari gambar tersebut dapat dijelaskan bahwa nilai-nilai prediksi dari taksiran model regresi non-linear yang diperoleh relatif baik, karena sudah mengikuti pola yang ada pada data. Kondisi ini berbeda dengan yang dideskripsikan oleh nilai-nilai residual model di Gambar 14.2b. Grafik residual tersebut menunjukkan bahwa residual model belum menunjukkan pola yang random, melainkan pola yang cenderung mengandung sifat autokorelasi atau berkaitan dengan residual sebelum atau sesudahnya.



Gambar 14.2. Plot perbandingan kesesuaian prediksi dan evaluasi residual model

14.2. Perintah `nls` dan `SSasymptOrig` untuk estimasi model non-linear

Ada beberapa perintah di paket **R** yang disediakan untuk menjalankan model regresi non-linear, antara lain `nls` dan `SSasymptOrig`. Contoh diatas merupakan salah satu aplikasi dari perintah `nls` pada suatu data real. Pada bagian ini akan diberikan rangkuman tentang perintah `nls` dan `SSasymptOrig`, khususnya yang berkaitan dengan argumen-argumen yang dapat ditampilkan. Secara umum penggunaan perintah `nls` dan keterangan argumen yang disediakan adalah sebagai berikut.

```
nls(formula, data, start, control, algorithm, trace, subset, weights,
    na.action, model, lower, upper, ...)
```

Argumen	Keterangan
formula	a nonlinear model formula including variables and parameters. Will be coerced to a formula if necessary.
data	an optional data frame in which to evaluate the variables in <i>formula</i> and <i>weights</i> . Can also be a list or an environment, but not a matrix.
start	a named list or named numeric vector of starting estimates. When <i>start</i> is missing, a very cheap guess for <i>start</i> is tried (if <i>algorithm</i> != "plinear").
control	an optional list of control settings. See nls.control for the names of the settable control values and their effect.
algorithm	character string specifying the algorithm to use. The default algorithm is a Gauss-Newton algorithm. Other possible values are "plinear" for the Golub-Pereyra algorithm for partially linear least-squares models and "port" for the 'nl2sol' algorithm from the Port package.
trace	logical value indicating if a trace of the iteration progress should be printed. Default is <i>FALSE</i> . If <i>TRUE</i> the residual (weighted) sum-of-squares and the parameter values are printed at the conclusion of each iteration. When the "plinear" algorithm is used, the conditional estimates of the linear parameters are printed after the nonlinear parameters. When the "port" algorithm is used the objective function value printed is half the residual (weighted) sum-of-squares.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional numeric vector of (fixed) weights. When present, the objective function is weighted least squares.
na.action	a function which indicates what should happen when the data contain <i>NA</i> s. The default is set by the <i>na.action</i> setting of <i>options</i> , and is na.fail if that is unset. The 'factory-fresh' default is na.omit . Value na.exclude can be useful.
model	logical. If true, the model frame is returned as part of the object. Default is <i>FALSE</i> .
lower, upper	vectors of lower and upper bounds, replicated to be as long as <i>start</i> . If unspecified, all parameters are assumed to be unconstrained. Bounds can only be used with the "port" algorithm. They are ignored, with a warning, if given for other algorithms.
...	Additional optional arguments. None are used at present.

SSasympOrig merupakan suatu perintah untuk aplikasi regresi non-linear dengan fungsi yang spesifik, yaitu

$$y = \frac{Asym}{1 - \exp(-\exp(lrc) * input)} + \varepsilon ,$$

dengan *input* adalah suatu vektor kovariat, dan *Asym* dan *lrc* merupakan parameter-parameter model. Secara umum penggunaan perintah **SSasympOrig** dan keterangan argumen yang disediakan adalah sebagai berikut.

SSasympOrig(input, Asym, lrc)

Argumen	Keterangan
input	<i>a numeric vector of values at which to evaluate the model.</i>
Asym	<i>a numeric parameter representing the horizontal asymptote.</i>
lrc	<i>a numeric parameter representing the natural logarithm of the rate constant.</i>

Misalkan perintah **SSasympOrig** akan diaplikasikan pada data **BOD** yang sudah tersedia di **R**. Data ini terdiri dari dua variabel, yaitu *Time* dan *demand*. Berikut ini adalah **script** untuk aplikasi perintah **SSasympOrig** dan beberapa output berkaitan dengan taksiran model regresi non-linear dan plot prediksi yang dihasilkan.

```
> BOD
      Time demand
1         1    8.3
2         2   10.3
3         3   19.0
4         4   16.0
5         5   15.6
6         7   19.8

> fm <- nls(demand ~ SSasympOrig(Time, A, lrc), data = BOD)
> # fm <- nls(demand ~ A*(1 - exp(-exp(lrc)*Time)), data=BOD,
  start = list(A = 1, lrc=0.1), trace=TRUE)
```

```

> summary(fm)

Formula: demand ~ A * (1 - exp(-exp(lrc) * Time))

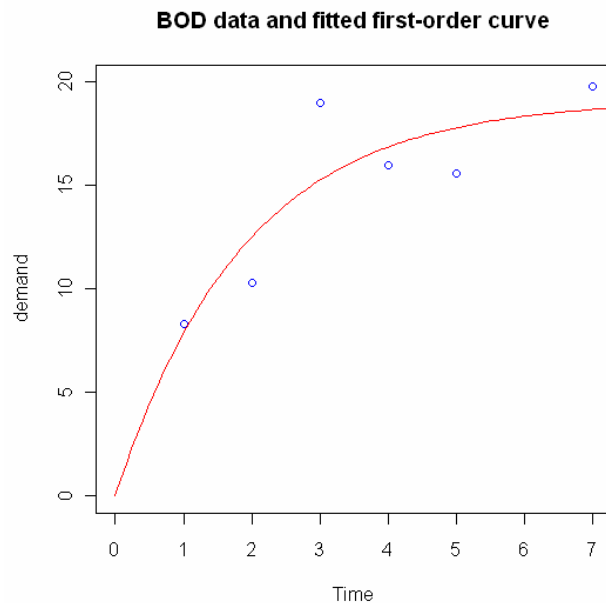
Parameters:
      Estimate Std. Error t value Pr(>|t|)
A    19.1426    2.4959   7.670  0.00155 **
lrc   -0.6328    0.3824  -1.655   0.17328
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.549 on 4 degrees of freedom

Number of iterations to convergence: 7
Achieved convergence tolerance: 5.54e-07

> predict(fm)      # fitted values at observed times
[1] 7.887449 12.524977 15.251673 16.854870 17.797490 18.677580
> plot(demand ~ Time, data = BOD, col = 4,
+      main = "BOD data and fitted first-order curve",
+      xlim = c(0,7), ylim = c(0, 20) )
> tt <- seq(0, 8, length = 101)
> lines(tt, predict(fm, list(Time = tt)), col=2)

```



Gambar 14.3. Plot perbandingan kesesuaian prediksi pada data **BOD**

14.3. Uji Deteksi Hubungan Non-linear

Paket **R** menyediakan beberapa uji untuk mendeteksi hubungan non-linear atau *non-linearity test* antar variabel, baik pada model regresi ataupun analisis runtun waktu. Pada bagian ini pembahasan difokuskan pada deteksi non-linearitas pada model regresi, khususnya Uji **Ramsey's RESET**, Uji **White**, dan Uji **Terasvirta**. Berikut adalah penjelasan untuk masing-masing uji non-linearitas tersebut.

14.3.1. Uji Ramsey's RESET

Teori berkaitan dengan Uji **Ramsey's RESET** ini secara lengkap dapat dilihat di Ramsey (1969), dan Gujarati (1996). Dalam **R**, Uji **Ramsey's RESET** disediakan pada library **lmtest** dengan perintah **resettest**. Secara umum, penggunaan perintah **resettest** yang disediakan di library **lmtest** adalah sebagai berikut.

```
resettest(formula, power = 2:3, type = c("fitted", "regressor",
"princomp"), data = list())
```

Keterangan lengkap tentang argumen yang dapat digunakan untuk perintah **resettest** adalah sebagai berikut.

Argumen	Keterangan
formula	<i>a symbolic description for the model to be tested (or a fitted "lm" object).</i>
power	<i>integers. A vector of positive integers indicating the powers of the variables that should be included. By default, the test is for quadratic or cubic influence of the fitted response.</i>
type	<i>a string indicating whether powers of the fitted response, the regressor variables (factors are left out), or the first principal component of the regressor matrix should be included in the extended model.</i>
data	<i>an optional data frame containing the variables in the model. By default the variables are taken from the environment which resettest is called from.</i>

Penjelasan tentang argumen **power** dengan isian **integers** adalah suatu fasilitas tentang suatu bilangan bulat positif yang mengindikasikan pangkat dari variabel-variabel yang akan ditambahkan dalam pengujian non-linearitas model. Secara **default**, uji ini adalah untuk pengaruh kuadrat (pangkat 2) atau kubik (pangkat 3) dari taksiran variabel respon. Sedangkan argumen **type** dengan tiga pilihan isian yang harus ditambahkan dalam model untuk pengujian non-linearitas, yaitu *the fitted response* (taksiran variabel respon), *the regressor variables* (variabel prediktor), dan *the first principal component* (komponen utama pertama) dari matriks regressor. Berikut adalah ringkasan prosedur Uji **Ramsey's RESET**, dengan **power=3**, dan pilihan taksiran variabel respon (*the fitted response*) sebagai prediktor tambahan.

- (i). Regresikan y_t pada $1, x_1, \dots, x_p$ dan hitung nilai-nilai taksiran variabel respon \hat{y}_t , yaitu

$$\hat{y}_t = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p.$$

Hitung koefisien determinasi dari regresi, yaitu R^2 , dan selanjutnya notasikan dengan R_{old}^2 .

- (ii). Regresikan y_t pada $1, x_1, \dots, x_p$ dan 2 prediktor tambahan, yaitu \hat{y}_t^2 dan \hat{y}_t^3 , dengan model

$$y_t = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p + \alpha_1 \hat{y}_t^2 + \alpha_2 \hat{y}_t^3.$$

Kemudian hitung koefisien determinasi dari regresi ini, yaitu R^2 , dan notasikan dengan R_{new}^2 .

- (iii). Hitung nilai uji F , yaitu

$$F = \frac{(R_{new}^2 - R_{old}^2) / m}{(1 - R_{new}^2) / (n - p - 1 - m)},$$

dengan m : banyaknya prediktor tambahan (dalam hal ini 2, suku kuadrat dan kubik),

p : banyaknya prediktor awal, dan

n : jumlah pengamatan yang digunakan.

Dibawah hipotesis linearitas, nilai uji F ini mendekati distribusi F dengan derajat bebas m dan $(n - p - 1 - m)$.

Secara umum, argumen **type** akan memberikan perbedaan model pada tahap (ii) dari prosedur uji diatas. Jika pilihan **type** adalah variabel *regressor*, maka m prediktor tambahan dalam tahap (ii) adalah suku kuadrat dan/atau kubik dari variabel *regressor*, sesuai dengan pilihan **power** yang digunakan.

Dalam Uji **Ramsey's RESET** ini, bentuk umum model yang menjelaskan hubungan antara variabel independen (prediktor) dengan variabel dependen (respon) dapat ditulis dalam

$$Y = f(X) + \varepsilon .$$

Hipotesis pengujian yang digunakan dalam uji non-linearitas ini adalah :

H_0 : $f(X)$ adalah fungsi linear dalam X atau model linear

H_1 : $f(X)$ adalah fungsi non-linear dalam X atau model non-linear .

H_0 ditolak yang berarti model non-linear adalah yang sesuai, jika nilai uji F memenuhi daerah penolakan yaitu

$$F > F_{\alpha; (df_1=m, df_2=n-p-1-m)} \quad \text{atau} \quad p\text{-value} < \alpha .$$

Misalkan perintah **resettest** akan diaplikasikan untuk pengujian dua macam data simulasi, yaitu dari model nonlinear (Y_1) dan model linear (Y_2). Bentuk matematis dari model pada data simulasi adalah

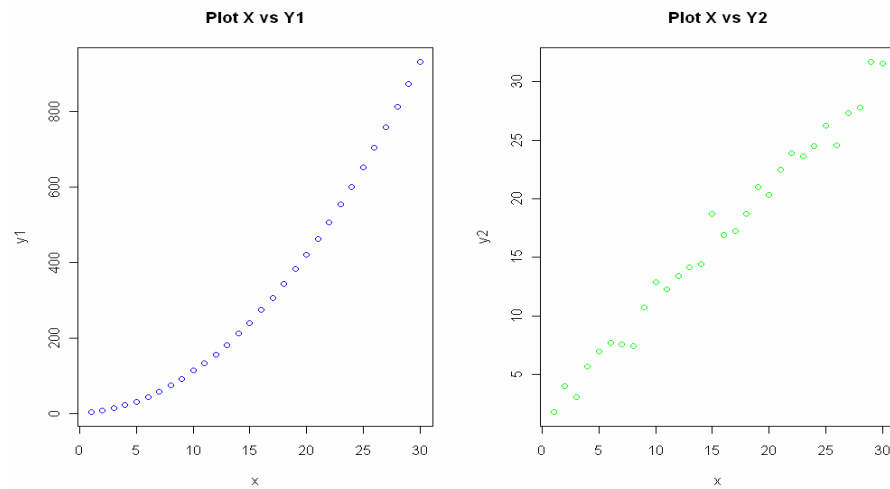
$$(i). \quad Y_1 = 1 + X + X^2 + \varepsilon ,$$

$$(ii). \quad Y_2 = 1 + X + \varepsilon ,$$

dengan $X = \{1, 2, \dots, 30\}$ dan $\varepsilon \sim N(0, 1)$. Berikut ini adalah *script* untuk membangkitkan data dan membuat plot yang menggambarkan hubungan antara X dengan Y_1 dan Y_2 .

```
> x <- c(1:30)
> y1 <- 1 + x + x^2 + rnorm(30)
> y2 <- 1 + x + rnorm(30)
> win.graph()
> par(mfrow=c(1,2))
> plot(x,y1,main="Plot X vs Y1",col="blue")
> plot(x,y2,main="Plot X vs Y2",col="green")
```

Hasil dari plot yang menggambarkan hubungan antara X dengan Y_1 dan Y_2 dapat dilihat pada Gambar 14.4.



Gambar 14.4. Plot antara X dengan Y_1 , dan X dengan Y_2

Selanjutnya, aplikasi Uji **Ramsey's RESET** pada kedua pasangan data dapat dilakukan dengan menggunakan perintah berikut ini.

```
> library(lmtest) # Aktifkan terlebih dahulu
> resettest(y1 ~ x, power=2, type="regressor")
> resettest(y2 ~ x, power=2, type="regressor")
```

Hasil dari perintah Uji **Ramsey's RESET** untuk kedua pasangan data di atas adalah sebagai berikut.

```
> resettest(y1 ~ x, power=2, type="regressor")

RESET test
data: y1 ~ x
RESET = 170757.1, df1 = 1, df2 = 27, p-value < 2.2e-16

> resettest(y2 ~ x, power=2, type="regressor")

RESET test
data: y2 ~ x
RESET = 0.4505, df1 = 1, df2 = 27, p-value = 0.5078
```

Berdasarkan output diatas dapat disimpulkan bahwa ada hubungan non-linear antara X dengan Y_1 . Hal ini ditunjukkan oleh p -value (**2.2e-16**) yang lebih kecil dari $\alpha=5\%$. Sehingga model non-linear adalah model yang sesuai untuk menjelaskan hubungan antara X dengan Y_1 . Sebaliknya, output tersebut menunjukkan bahwa tidak ada hubungan non-linear antara X dengan Y_2 , dan ini dijelaskan oleh p -value (**0.5078**) yang lebih besar dari $\alpha=5\%$. Dengan demikian, model linear adalah model yang sesuai untuk menjelaskan hubungan antara X dengan Y_2 .

Sebagai tambahan, perhatikan kembali data simulasi pada model pertama, yaitu model nonlinear (Y_1) yang merupakan model kuadrat. Jika didefinisikan $X_1 = X$, $X_2 = X^2$ dan keduanya dimasukkan sebagai variabel prediktor (regressor) pada Uji **Ramsey's RESET**, maka akan diperoleh hasil seperti berikut ini.

```
> x1 <- x
> x2 <- x^2
> resettest(y1 ~ x1+x2, power=2, type="regressor")
```

RESET test

data: $y_1 \sim x_1 + x_2$

RESET = 0.2263, df1 = 2, df2 = 25, p-value = 0.799

Hasil output ini menunjukkan bahwa model yang sesuai untuk menjelaskan hubungan antara X_1 dan X_2 dengan Y_1 adalah model linear. Hal ini ditunjukkan oleh p -value (**0.799**) yang lebih besar dari $\alpha=5\%$. Dengan demikian dapat disimpulkan bahwa tidak ada hubungan non-linear antara X_1 dan X_2 dengan Y_1 .

14.3.2. Uji White

Uji **White** adalah uji deteksi nonlinearitas yang dikembangkan dari model neural network yang dikemukakan oleh White (1989). Uji ini termasuk dalam kelompok uji tipe *Lagrange Multiplier* (LM). Secara lengkap teori berkaitan dengan uji White ini dapat dilihat di White (1989) dan Lee dkk. (1993). Pada paket **R**, Uji **White** disediakan pada library **tseries** dengan perintah **white.test**.

Berikut ini adalah penjelasan singkat tentang uji deteksi non-linearitas tipe *Lagrange Multiplier*. Misalkan I_t adalah suatu himpunan informasi yang didefinisikan

$$I_t = \{x_{1t}, x_{2t}, \dots, x_{p_t}\}$$

dan menyatakan semua variabel-variabel eksogen x_t yang digunakan dalam I_t oleh w_t .

Proses pemodelan adalah mendapatkan suatu pendekatan yang baik untuk $f(w_t)$ sedemikian hingga

$$E[y_t | I_t] = f(w_t) .$$

Strategi pemodelan pada model statistik yang nonlinear dapat dilakukan dalam dua tahap, yaitu (i) uji linearitas untuk y_t dengan menggunakan informasi I_t , dan (ii) jika linearitas ditolak, gunakan beberapa model parametrik alternatif, nonparametrik, dan atau semiparametrik.

Perhatikan suatu model nonlinear yang secara matematis ditulis dalam bentuk sebagai berikut

$$y_t = \varphi(\gamma'w_t) + \beta'w_t + u_t \quad (14.1)$$

dengan $u_t \sim \text{IIDN}(0, \sigma^2)$, $w_t = (1, \tilde{w}_t')'$, $\tilde{w}_t = (x_{1t}, \dots, x_{pt})'$, $\beta = (\beta_0, \beta_1, \dots, \beta_p)'$, $\gamma = (\gamma_0, \tilde{\gamma}')'$, dan $\tilde{\gamma} = (\gamma_1, \dots, \gamma_p)'$. Misal diberikan

$$\varphi(\gamma'w_t) = \theta_0 \psi(\gamma'w_t) , \quad (14.2)$$

dengan

$$\psi(\gamma'w_t) = \{1 + \exp(-\gamma'w_t)\}^{-1} - \frac{1}{2} .$$

Dengan demikian persamaan (14.1) dapat diinterpretasikan sebagai suatu model regresi yang nonlinear dengan konstanta $\beta_0 + \theta_0 \psi(\gamma'w_t)$.

Model (14.1) adalah suatu kasus khusus dari model neural network satu layer tersembunyi (*hidden layer*),

$$y_t = \beta'w_t + \sum_{j=1}^q \theta_{0j} \{\psi(\gamma_j'w_t) - \frac{1}{2}\} + u_t . \quad (14.3)$$

Secara visual, arsitektur dari model neural network ini dapat dilustrasikan seperti pada Gambar 14.5.

Perhatikan persamaan (14.1) dengan (14.2) dan suatu uji hipotesis bahwa y_t adalah linear, yaitu $y_t = \beta'w_t + u_t$. Hipotesis nol dapat didefinisikan sebagai $H_0 : \theta_0 = 0$. Dalam model (14.3) hipotesis

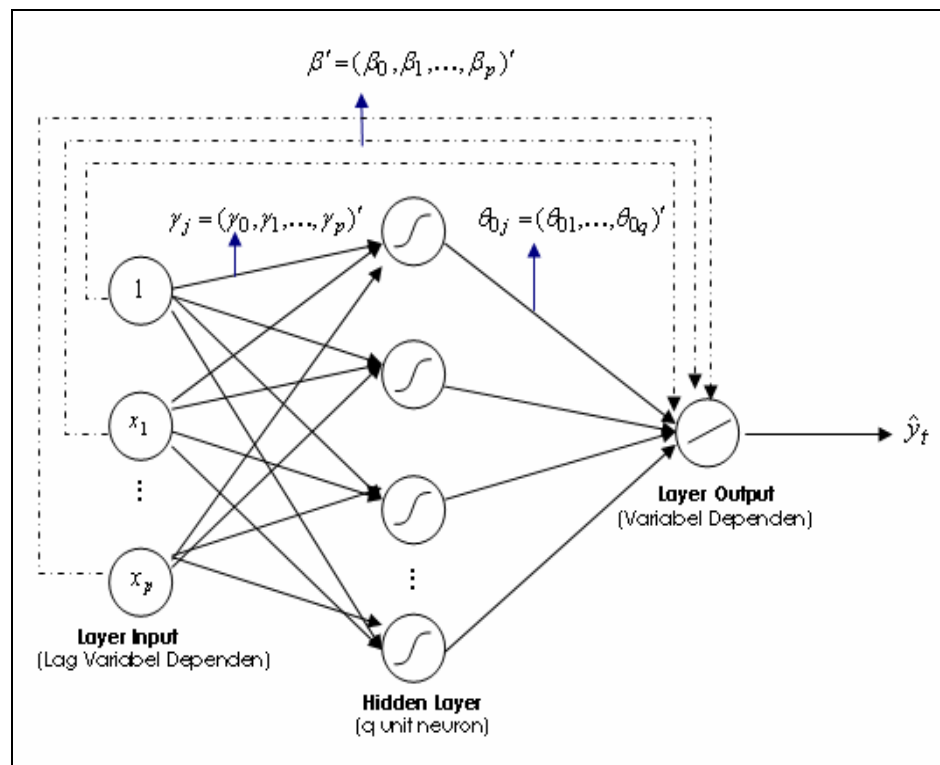
$$H_0 : \theta_{01} = \theta_{02} = \dots = \theta_{0q} = 0$$

disebut hipotesis linearitas dari uji neural network melawan nonlinearitas yang terabaikan (White, 1989; Lee dkk., 1993). Permasalahan identifikasi di atas diselesaikan dengan menetapkan nilai-nilai dari vektor $\gamma_1, \dots, \gamma_q$ sehingga nilai-nilai dari $\psi(\gamma_j'w_t)$ dapat dihitung. Hal ini dilakukan melalui penentuan vektor-vektor itu secara random dari suatu distribusi yang mungkin, dimana Lee dkk. (1993) menggunakan suatu distribusi uniform.

Karena variabel-variabel $\psi(\gamma_j' w_t)$ dimungkinkan sangat berkorelasi, Lee dkk. (1993) menerapkan suatu transformasi komponen utama menjadi

$$\bar{\psi}_t = [\psi(\gamma_1' w_t), \dots, \psi(\gamma_q' w_t)]'$$

dan menggunakan dua komponen utama yang ortonormal kedalam bagian linear dari model pada regresi tambahan untuk uji linearitas.



Gambar 14.5. Arsitektur model neural network satu layer tersembunyi pada persamaan (14.3)

Implementasi praktis dari uji linearitas yang merupakan tipe LM dengan sampling random yang dikenalkan oleh Lee dkk. (1993) yang selanjutnya dikenal dengan Uji **White** ini dapat dilakukan melalui dua statistik uji, yaitu uji χ^2 dan uji F . Berikut ini adalah penjelasan lengkap tentang prosedur untuk mendapatkan nilai uji χ^2 dan uji F pada Uji **White**.

▪ Prosedur untuk mendapatkan nilai uji χ^2 pada Uji **White**

- (i). Regresikan y_t pada $1, x_1, \dots, x_p$ dan hitung nilai-nilai residual \hat{u}_t .
- (ii). Regresikan \hat{u}_t pada $1, x_1, \dots, x_p$ dan m prediktor tambahan, dan kemudian hitung koefisien determinasi dari regresi R^2 . Dalam uji ini, m prediktor tambahan ini adalah nilai-nilai dari $\psi(\gamma'_j w_t)$ hasil dari suatu transformasi komponen utama.
- (iii). Hitung $\chi^2 = nR^2$, dengan n adalah jumlah pengamatan yang digunakan.

Dibawah hipotesis linearitas, χ^2 mendekati distribusi $\chi^2(m)$. Kajian teoritis berkaitan dengan pendekatan asimtotis $nR^2 \xrightarrow{d} \chi^2$ dapat dilihat pada White (1989).

▪ Prosedur untuk mendapatkan nilai uji F pada Uji **White**

- (i). Regresikan y_t pada $1, x_1, \dots, x_p$ dan hitung nilai-nilai residual \hat{u}_t dan hitung jumlah kuadrat residual $SSR_0 = \sum \hat{u}_t^2$.
- (ii). Regresikan \hat{u}_t pada $1, x_1, \dots, x_p$ dan m prediktor tambahan (seperti yang dijelaskan diatas), dan kemudian hitung residual \hat{v}_t dan jumlah kuadrat residual $SSR_1 = \sum \hat{v}_t^2$.

(iii). Hitung
$$F = \frac{(SSR_0 - SSR_1)/m}{SSR_1/(n - p - 1 - m)},$$

dengan n adalah jumlah pengamatan yang digunakan.

Dibawah hipotesis linearitas, nilai uji F ini mendekati distribusi F dengan derajat bebas m dan $(n - p - 1 - m)$.

Secara umum, ada dua macam penggunaan perintah **white.test** yang disediakan di library **tseries** untuk implementasi Uji **White**, yaitu untuk permasalahan analisis regresi dan analisis runtun waktu. Berikut ini adalah cara penggunaan uji **White** tersebut.

```
## Default method: untuk analisis regresi
white.test(x, y, qstar = 2, q = 10, range = 4,
           type = c("Chisq", "F"), scale = TRUE, ...)

## Uji White untuk analisis runtun waktu
white.test(x, lag = 1, qstar = 2, q = 10, range = 4,
           type = c("Chisq", "F"), scale = TRUE, ...)
```

Keterangan tentang argumen yang dapat digunakan untuk perintah **white.test** adalah sebagai berikut.

Argumen	Keterangan
x	<i>A numeric vector, matrix, or time series.</i>
y	<i>A numeric vector.</i>
lag	<i>an integer which specifies the model order in terms of lags.</i>
q	<i>an integer representing the number of phantom hidden units used to compute the test statistic.</i>
qstar	<i>the test is conducted using qstar principal components of the phantom hidden units. The first principal component is omitted since in most cases it appears to be collinear with the input vector of lagged variables. This strategy preserves power while still conserving degrees of freedom.</i>
range	<i>the input to hidden unit weights are initialized uniformly over $[-range/2, range/2]$.</i>
type	<i>A string indicating whether the Chi-Squared test or the F-test is computed. Valid types are "Chisq" and "F".</i>
scale	<i>A logical indicating whether the data should be scaled before computing the test statistic. The default arguments to scale are used.</i>
...	<i>further arguments to be passed from or to methods.</i>

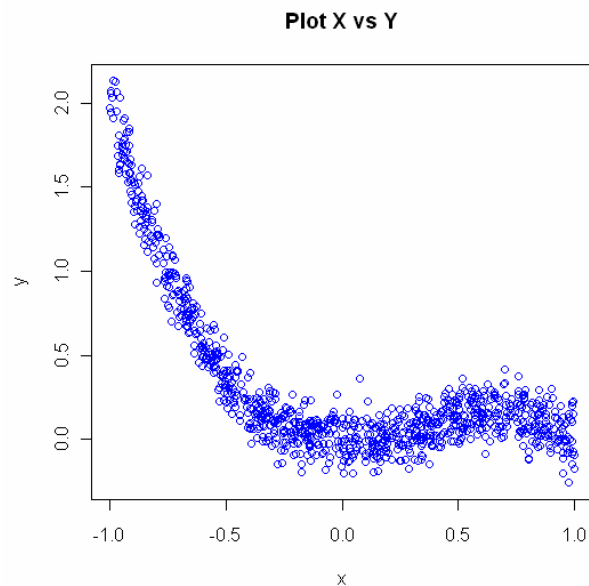
Misalkan perintah **white.test** akan diaplikasikan untuk pengujian data simulasi, yaitu dari model nonlinear dengan bentuk matematis sebagai berikut

$$Y = X^2 - X^3 + \varepsilon ,$$

dengan $X \sim U(-1,1)$ dan $\varepsilon \sim N(0,0.1)$. Berikut ini adalah *script* untuk membangkitkan data dan membuat plot yang menggambarkan hubungan antara X dengan Y .

```
> n <- 1000  
> x <- runif(1000, -1, 1) # Non-linear in ``mean'' regression  
> y <- x^2 - x^3 + 0.1*rnorm(x)  
> plot(x,y,main="Plot X vs Y",col="blue")
```

Hasil dari plot yang menggambarkan hubungan antara X dengan Y dapat dilihat pada Gambar 14.6 berikut ini.



Gambar 14.6. Plot yang menggambarkan hubungan antara X dengan Y

Berdasarkan plot ini dapat dijelaskan bahwa hubungan antara X dengan Y adalah non-linear. Berikut ini adalah aplikasi Uji **White** untuk deteksi non-linearitas pada pasangan data hasil simulasi tersebut.


```

> library(tseries) # Aktifkan terlebih dahulu
> white.test(x, y) # Uji Chi-square
> white.test(x, y, type = c("F")) # Uji F

> ## Is the polynomial of order 2 misspecified?
> white.test(cbind(x,x^2,x^3), y)

```

Dalam kasus ini, secara umum hipotesis pengujian yang digunakan dalam uji non-linearitas ini adalah :

H_0 : $f(X)$ adalah fungsi linear dalam X atau model linear

H_1 : $f(X)$ adalah fungsi non-linear dalam X atau model non-linear .

Hasil dari perintah Uji **Ramsey's RESET** untuk kedua pasangan data di atas adalah sebagai berikut.

```

> white.test(x, y)

      White Neural Network Test

data: x and y
X-squared = 2462.076, df = 2, p-value < 2.2e-16

> white.test(x, y, type = c("F"))

      White Neural Network Test

data: x and y
F = 4536.783, df1 = 2, df2 = 997, p-value < 2.2e-16

```

Berdasarkan output ini dapat disimpulkan bahwa ada hubungan non-linear antara X dengan Y , baik dengan uji Chi-square atau uji F. Hal ini ditunjukkan oleh besarnya *p-value* (**2.2e-16** baik pada uji Chi-square maupun uji F) yang lebih kecil dari $\alpha=5\%$. Dengan demikian, model non-linear adalah model yang sesuai untuk menjelaskan hubungan antara X dengan Y .

Sebagai tambahan, perhatikan kembali data simulasi pada model ini, yaitu Y merupakan model nonlinear yang merupakan model kubik. Jika didefinisikan $X_1 = X$, $X_2 = X^2$, $X_3 = X^3$ dan ketiganya dimasukkan sebagai variabel prediktor (regressor) pada Uji **White**, maka akan diperoleh hasil seperti output berikut ini.

```
> x1 <- x
> x2 <- x^2
> x3 <- x^3
> x.all <- cbind(x1,x2,x3)
> white.test(x.all, y)
```

White Neural Network Test

data: x.all and y

X-squared = 1.2287, df = 2, p-value = 0.541

```
> # Uji White dengan uji F
> white.test x.all, y, type = c("F"))
```

White Neural Network Test

data: x.all and y

F = 0.1956, df1 = 2, df2 = 995, p-value = 0.8224

Berbeda dengan kesimpulan sebelumnya, hasil output ini menunjukkan bahwa model yang sesuai untuk menjelaskan hubungan antara X_1 , X_2 dan X_3 dengan Y adalah model linear. Hal ini ditunjukkan oleh p -value (**0.541** pada uji Chi-square, dan **0.8224** pada uji F) yang lebih besar dari $\alpha=5\%$. Dengan demikian dapat disimpulkan bahwa tidak ada hubungan non-linear antara X_1 , X_2 dan X_3 dengan Y .

14.3.3. Uji Terasvirta

Seperti Uji **White** sebelumnya, Uji **Terasvirta** adalah uji deteksi nonlinearitas yang juga dikembangkan dari model neural network dan termasuk dalam kelompok uji tipe *Lagrange Multiplier* (LM). Secara lengkap teori berkaitan dengan uji Terasvirta ini dapat dilihat di Terasvirta dkk. (1993). Uji **Terasvirta** ini adalah uji tipe *Lagrange Multiplier* yang dikembangkan dengan ekspansi Taylor. Pada akhirnya, perbedaan utama dengan Uji **White** terletak pada tahap kedua prosedur uji, khususnya tentang m prediktor tambahan yang dimasukkan dalam model pengujian. Dalam Uji **Terasvirta** ini, m prediktor tambahan yang digunakan adalah suku kuadratik dan kubik yang merupakan hasil dari pendekatan ekspansi Taylor.

Paket **R** menyediakan fasilitas untuk Uji **Terasvirta** pada library **tseries** dengan perintah **teravirta.test**. Seperti pada Uji **White**, ada dua macam penggunaan perintah **teravirta.test** yang disediakan di library **tseries** untuk implementasi Uji **Terasvirta**, yaitu untuk permasalahan analisis regresi dan analisis runtun waktu. Berikut ini adalah cara penggunaan uji **Terasvirta** dengan paket **R**.

```
## Default method: untuk analisis regresi
terasvirta.test(x, y, type = c("Chisq", "F"),
               scale = TRUE, ...)

## Uji Terasvirta untuk analisis runtun waktu
terasvirta.test(x, lag = 1, type = c("Chisq", "F"),
               scale = TRUE, ...)
```

Keterangan tentang argumen yang dapat digunakan untuk perintah **terasvirta.test** adalah sebagai berikut.

Argumen	Keterangan
x	<i>a numeric vector, matrix, or time series.</i>
y	<i>a numeric vector.</i>
lag	<i>an integer which specifies the model order in terms of lags.</i>
type	<i>a string indicating whether the Chi-Squared test or the F-test is computed. Valid types are "Chisq" and "F".</i>
scale	<i>a logical indicating whether the data should be scaled before computing the test statistic. The default arguments to scale are used.</i>
...	<i>further arguments to be passed from or to methods.</i>

Misalkan perintah **terasvirta.test** akan diaplikasikan untuk pengujian data simulasi seperti pada Uji **White** sebelumnya, yaitu dari model nonlinear dengan bentuk matematis sebagai berikut

$$Y = X^2 - X^3 + \varepsilon ,$$

dengan $X \sim U(-1,1)$ dan $\varepsilon \sim N(0,0.1)$. Seperti pada bagian sebelumnya, hipotesis pengujian yang digunakan adalah

H_0 : $f(X)$ adalah fungsi linear dalam X atau model linear

H_1 : $f(X)$ adalah fungsi non-linear dalam X atau model non-linear .

Berikut ini adalah *script* dan hasil implementasi Uji **Terasvirta** untuk evaluasi hubungan antara X dengan Y .

```
> terasvirta.test(x, y)

Teraesvirta Neural Network Test

data: x and y
X-squared = 2476.359, df = 2, p-value < 2.2e-16

> terasvirta.test(x, y, type = c("F"))

Teraesvirta Neural Network Test

data: x and y
F = 5432.583, df1 = 2, df2 = 997, p-value < 2.2e-16

> terasvirta.test(x.all, y)

Teraesvirta Neural Network Test

data: x.all and y
X-squared = 21.7509, df = 16, p-value = 0.1514

> terasvirta.test(x.all, y, type = c("F"))

Teraesvirta Neural Network Test

data: x.all and y
F = 1.3482, df1 = 16, df2 = 981, p-value = 0.1605
```

Berdasarkan output dari dua perintah pertama dapat disimpulkan bahwa ada hubungan non-linear antara X dengan Y , baik dengan uji Chi-square atau uji F. Hal ini ditunjukkan oleh besarnya *p-value* (**2.2e-16** baik pada uji Chi-square maupun uji F) yang lebih kecil dari $\alpha=5\%$. Dengan demikian, model non-linear adalah model yang sesuai untuk menjelaskan hubungan antara X dengan Y , dan hal ini sama dengan hasil yang diperoleh pada Uji **White**.

Sedangkan hasil output dari dua perintah terakhir menunjukkan bahwa model yang sesuai untuk menjelaskan hubungan antara $X_1 = X$, $X_2 = X^2$ dan $X_3 = X^3$ dengan Y adalah model linear. Hal ini ditunjukkan oleh *p-value* (**0.1514** pada uji Chi-square, dan **0.1605** pada uji F) yang lebih besar dari $\alpha=5\%$. Hasil ini adalah sama dengan kesimpulan yang diperoleh pada Uji **White**, yaitu tidak ada hubungan non-linear antara X_1 , X_2 dan X_3 dengan Y .

BAB 15

PENGENALAN PEMROGRAMAN DALAM R

Paket **R** menyediakan fasilitas untuk membuat fungsi yang didefinisikan oleh *user* (*user-defined function*). Fasilitas ini memungkinkan *user* untuk membuat program analisis yang lebih fleksibel dengan menggunakan fungsi-fungsi *built-in* didalam **R**. Fungsi-fungsi *built-in* ini sudah diperkenalkan pada bab-bab sebelumnya. Pada bagian ini akan diberikan pengantar mengenai pemrograman dengan bahasa **R**.

15.1. Penulisan Fungsi

Salah satu tahap penting dalam pendefinisian fungsi-fungsi baru di **R** adalah cara penulisan fungsi tersebut. Secara umum, struktur penulisan fungsi di dalam **R** adalah sebagai berikut.

```
nama_fungsi = function(argumen dari fungsi)
{
..... isi dari fungsi
}
```

Penulisan fungsi ini dapat dilakukan melalui dua macam cara, yaitu melalui **R-Console** dan **R-Editor**. Sebagai ilustrasi, misalkan akan dibuat fungsi untuk menghitung rata-rata sekumpulan data. Dalam hal ini dianggap **R** tidak memiliki fungsi untuk menghitung nilai rata-rata. Akan tetapi **R** memiliki fungsi untuk menghitung jumlah (yakni **sum**), dan fungsi untuk menghitung panjang vektor (yakni **length**). Berikut ini adalah contoh fungsi rata-rata yang dituliskan pada **R-Console**.

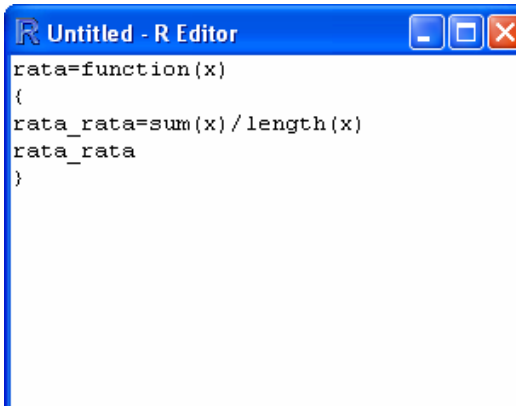
```
> rata=function(x)
+ {
+   rata_rata=sum(x)/length(x)
+   rata_rata
+ }
```

Dalam contoh ini, nama fungsi yang dibuat adalah **rata**, dan argumennya adalah **x** sebagai data yang akan dihitung rata-ratanya.

```
> x=c(2,1,3,4,5)
> rata(x)
[1] 3
```

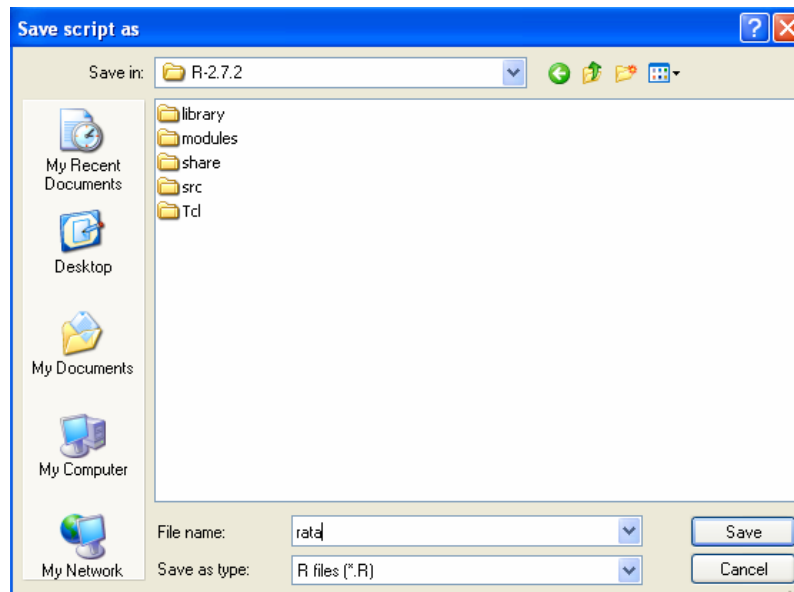
Sedangkan melalui **R-Editor**, pendefinisian fungsi baru dapat dikerjakan dengan langkah-langkah sebagai berikut.

- Pertama kali munculkan **R-Editor** dengan klik **File** pada **R-Console**, kemudian pilih **New script**. Selanjutnya ketik *script* fungsi yang akan dibuat pada **R-Editor** seperti berikut ini.



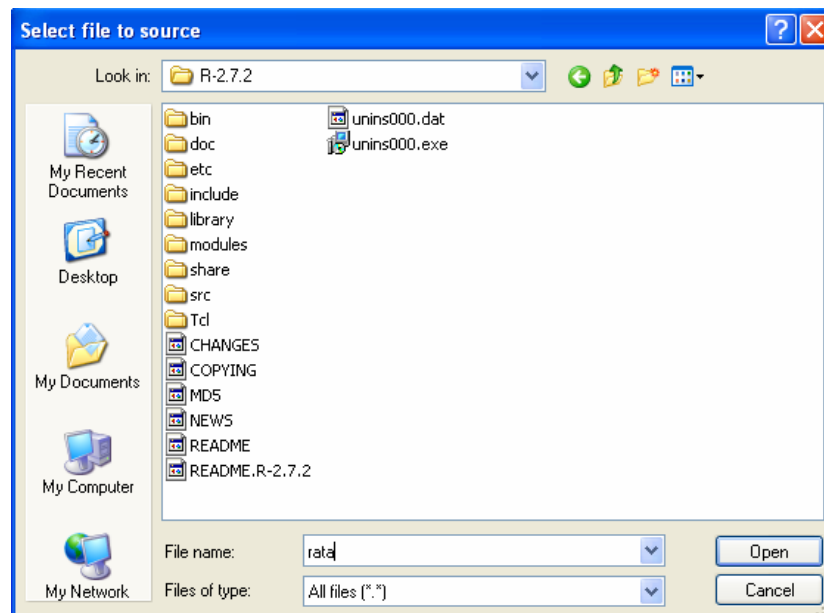
```
rata=function(x)
{
rata_rata=sum(x)/length(x)
rata_rata
}
```

- Apabila sudah selesai, simpan fungsi ini dengan memilih **File** pada **R-Console**, dan kemudian klik **Save as...**, sehingga diperoleh tampilan seperti berikut ini.



Sebagai catatan, pemberian nama boleh berbeda dengan nama fungsi yang didefinisikan. Akan tetapi untuk mempermudah mengingatnya disarankan untuk menggunakan nama file sama dengan nama fungsi. Dalam hal ini diberi nama *rata*.

- Selanjutnya agar fungsi dapat digunakan, maka didalam file ini harus dibuat *source*-nya. Proses ini dapat dilakukan dengan menggunakan menu **File**, kemudian klik **Source R code...** dan pilih file yang akan *disource*-kan seperti berikut.



- Klik **Open**, sehingga pada **R-Console** akan muncul *text* berikut ini.

```
> source("C:\\Program Files\\R\\R-2.7.2\\rata")
```

Text ini menunjukkan bahwa dalam fungsi yang didefinisikan tidak ada kesalahan perintah. Untuk mencoba fungsi ini, misalkan akan dihitung rata-rata dari data 3,1,2,4,5. Selanjutnya ketik perintah seperti berikut.

```
> x=c(3,1,2,4,5)
> rata(x)
[1] 3
```

Sampai di sini sudah bisa didefinisikan fungsi baru yang sederhana didalam **R**. Untuk pembuatan fungsi yang lebih kompleks, diperlukan pengetahuan tentang *type data* dan perintah-perintah dalam pemrograman.

15.2. Type Data dan Operator

Pada Bab 3 telah dibahas tentang beberapa *type* data pada **R**. Pengetahuan akan *type* data ini sangat diperlukan dalam pendefinisian fungsi baru. Secara umum ada *type* data dasar, yaitu **numeric**, **character** atau **string**, dan **logika**. Sedangkan *type* data yang terstruktur, yang terdiri dari kombinasi *type* data dasar, adalah **array**. **Array** yang terdiri dari satu dimensi dinamakan vektor, sedangkan yang terdiri dari dua dimensi atau lebih dinamakan matrik. Bagaimana mendefinisikan data-data *type* ini secara lengkap sudah dibahas dalam Bab 3.

Hal lain yang harus diingat selain *type* data adalah **operator**. **Operator** digunakan untuk operasi matematik atau manipulasi data. Dalam pendefinisian fungsi baru atau secara umum dalam pemrograman seringkali memerlukan kombinasi penggunaan berbagai macam **operator**. Berikut ini ringkasan **operator** yang biasa digunakan dalam operasi matematik.

▪ Operator Aritmetika

+	penjumlahan
-	pengurangan
*	perkalian
/	pembagian
^	pangkat

▪ Operator Logika

<	kurang dari
<=	kurang dari atau sama dengan
>	lebih dari
>=	lebih dari atau sama dengan
==	sama dengan
!=	tidak sama dengan
&	dan (<i>and</i>)
	atau (<i>or</i>)
!	tidak (<i>negasi</i>)

Selain **operator** diatas, dalam Bab 3 telah pula dibuat ringkasan **operator** untuk matrik seperti penjumlahan, pengurangan, perkalian, invers, determinan dan *eigen value*. Perintah-perintah ini sangat penting dalam mendefinisikan fungsi baru.

15.3. Control Flow di dalam R

Barisan perintah dalam **R** biasanya dieksekusi baris per baris. Dalam **R**, barisan perintah yang diletakkan didalam tanda kurung **{ }** sebagai satu group dipandang sebagai satu ekspresi tunggal. Barisan ekspresi yang dinyatakan dalam bentuk group ini juga di eksekusi baris per baris. Untuk mengatur proses eksekusi, diperlukan perintah-perintah *control flow*. Bentuk *control flow* ini akan banyak digunakan dalam menulis suatu fungsi yang di bentuk oleh *user*. Beberapa perintah *control flow* yang dikenal didalam **R** akan dibahas dalam bagian ini.

▪ Statemen if

Statemen ini mempunyai aturan penulisan

```
> if (kondisi) {ekspresi}
```

Perintah ini berarti jika (kondisi) bernilai benar, maka {**ekspresi**} dilaksanakan, jika (kondisi) bernilai salah, maka {**ekspresi**} tidak dilaksanakan. Berikut adalah contoh statemen **if**.

```
> if (2>3) x=c(1,2,3)
> x
Error: object "x" not found
```

Pada contoh ini, karena kondisi **(2>3)** bernilai **False**, maka ekspresi **x=c(1,2,3)** tidak dieksekusi. Sehingga pada saat nilai **x** ditampilkan, karena **x** tidak ada maka keluar pesan **Error: object "x" not found**. Selanjutnya perhatikan contoh statemen **if** berikut ini.

```
> if (2<3) x=c(1,2,3)
> x
[1] 1 2 3
```

Pada contoh ini, karena kondisi **(2<3)** bernilai benar (**True**), maka ekspresi **x=c(1,2,3)** dieksekusi, sehingga nilai-nilai data **x** dapat ditampilkan.

▪ **Statemen if else**

Statemen ini mempunyai aturan penulisan seperti berikut ini.

```
> if (kondisi) {ekspresi1} else {ekspresi2}
```

Perintah ini mempunyai arti jika (**kondisi**) bernilai benar maka **{ekspresi1}** yang dilaksanakan, dan jika bernilai salah maka **{ekspresi2}** yang dilaksanakan. Berikut contoh statemen **if else**.

```
> if (2>3) x=c(1,2,3) else x=c(4,5,6)
> x
[1] 4 5 6
```

▪ **Statemen for**

Statemen ini digunakan untuk perulangan. *Syntax* dasar dari fungsi ini adalah sebagai berikut.

```
> for (name in expr1) {expr2}
```

Pada contoh dibawah ini, statemen **for** akan digunakan untuk menghitung nilai faktorial.

```
> f=1
> for (i in 1:5)
+ {
+ f=f*i
+ }
> f
[1] 120
```

Pada contoh ini dihitung nilai $5! = 1.2.3.4.5 = 120$.

▪ **Fungsi break dan next**

Syntax dari fungsi ini adalah sebagai berikut.

break : stop dan keluar *loop* yang sedang dieksekusi.
next : stop iterasi yang sedang berjalan dan langsung mulai iterasi selanjutnya.

Command **next** dan **break** pada dasarnya berfungsi untuk mencegah kemungkinan adanya *infinitife loop* dalam suatu fungsi, seperti dalam suatu *loop repeat* (yang dalam

R tidak memiliki akhir eksekusi yang alami, sehingga mutlak diperlukan *stopping loop* dengan perintah **break** atau **next**) dan *while* (yang diperlukan untuk menghentikan perulangan atau *loop* di satu bagian dari ekspresi dalam *loop* tanpa melanjutkan ke ekspresi lain di dalam *loop* ini).

▪ Statemen return dan stop

Kedua bentuk statemen ini digunakan untuk menghentikan eksekusi dari suatu fungsi yang telah diakses dan kembali ke **R prompt**. *Syntax*nya adalah sebagai berikut.

return(expr) : stop fungsi yang sedang diakses atau dievaluasi dan munculkan output nilai dari **expr** di *prompt*.

stop(message) : digunakan untuk memberikan tanda adanya kesalahan dengan menghentikan evaluasi dari fungsi yang sedang diakses dan menampilkan *message* di *prompt* sebagai pesan kesalahan dan kembali ke **R prompt**.

Fungsi **return** tidak hanya membuat kita berhenti dari *loop* yang sedang dievaluasi, tetapi juga dari fungsi yang sedang diakses. Jadi berbeda dengan bentuk **break** atau **next** yang tidak menghentikan eksekusi fungsi.

▪ Statement repeat

Syntax dasar dari statemen ini adalah sebagai berikut.

```
> repeat {expr}
```

Perintah **repeat** pada **R** merupakan perintah yang mengakibatkan perulangan atau *looping* tiada henti. Oleh karena itu, penggunaan **repeat** memerlukan penambahan perintah untuk menghentikan perulangan. Perintah ini bisa menggunakan statemen **if** yang dikombinasikan dengan **break**. Berikut ini adalah contoh untuk menghitung **n!** dengan menggunakan **repeat**.

```
> f=1
> i=0
> repeat
+ {
+ i=i+1
+ f=f*i
+ if (i==5) break
+ }
> f
[1] 120
```

▪ Statement **while**

Statemen **while** merupakan statemen untuk perulangan, dengan *syntax* sebagai berikut.

```
> while (condition) expr
```

Sebagai contoh, akan dibuat perintah untuk menghitung nilai **n!**.

```
> f=1
> i=0
> while (i<5)
+ {
+ i=i+1
+ f=f*i
+ }
> f
[1] 120
```

15.4. Beberapa topik yang berhubungan dengan fungsi

Berikut ini adalah beberapa topik yang berhubungan dengan fungsi yang banyak digunakan dalam pemrogram **R**.

15.4.1. Argumen dari suatu fungsi

Didalam membuat argumen dari suatu fungsi ada beberapa hal yang diperhatikan, antara lain *optional* dan *required argument*.

a. Optional argument

Optional argument adalah argumen suatu fungsi yang dapat tidak diberikan nilainya ketika fungsi tersebut dipanggil. Untuk hal tersebut biasanya ada nilai *default* dari argumen itu yang tidak perlu didefinisikan nilainya pada saat dipanggil. Sebagai contoh, misalkan ingin dibangkitkan data berdistribusi normal sebanyak 10, dengan mean = 15 dan variansi = 9 dengan fungsi **rnorm** berikut ini.

```
datanormal=function(n=10,mean=15,variansi=9)
{
  data <- rnorm(n,mean,sqrt(variansi))
  data
}
```

Seluruh argumen dari fungsi **datanormal** diatas merupakan *optional argument* yakni secara *default* telah diberikan nilai dari masing-masing argumen. Setelah *script function* diketikkan pada **R-Console** atau pada **R-Editor**, panggil dengan perintah berikut.

```
> datanormal()  
[1] 12.486330 15.940636 16.792752 18.634015 13.500430 15.722838 10.076475  
[8] 9.990897 8.056660 15.627972
```

Sudah ditampilkan 10 bilangan random normal dengan mean 15 dan variansi 9. Argumen *optional* ini dapat diganti dengan memberikan spesifikasi nilai dari argumen tersebut. Sebagai contoh akan dibangkitkan 25 bilangan normal standar dengan fungsi diatas. Hal tersebut dapat dilakukan dengan memanggil fungsi **datanormal** dengan merubah nilai argumennya, yaitu

```
> datanormal(n=20, mean=0, variansi=1)
```

atau dapat juga dengan

```
> datanormal(20,0,1)
```

Perintah ini akan mengganti nilai *default* argumen *optional* pada fungsi **datanormal**, sehingga diperoleh hasil sebagai berikut ini.

```
[1] -1.14368017 0.13469967 2.59678936 1.47190479 -0.75785580 -1.73536802  
[7] -0.40579935 -1.21402239 -1.37558235 0.02939101 -1.51945577 -0.84322992  
[13] 1.69295140 -2.57290634 -0.52587204 1.64317426 -0.57395625 -0.23601621  
[19] 0.14749658 1.42066282
```

b. Required argument

Required argument adalah argumen-argumen yang harus diberikan atau dispesifikasi nilainya jika fungsi tersebut dipanggil. Sebagai contoh akan dilakukan modifikasi fungsi bangkitan normal diatas menjadi seperti berikut ini.

```
datanormal2 <-function(n, mean, variansi=9)  
{  
  data=rnorm(n, mean, sqrt(variansi))  
  data  
}
```

Dengan demikian argumen **n** dan **mean** merupakan *required argument*, sehingga harus diberi nilai ketika *user* memanggil fungsi bangkitan **datanormal2** melalui **R prompt**. Sebagai contoh, perhatikan perintah berikut ini.

```
> datanormal2(10,0)
```

Perintah ini akan membangkitkan 10 bilangan random normal dengan mean 0 dan variansi 9, dengan output sebagai berikut.

```
[1] -1.91467807 -1.51618959 -0.03283981 -1.54936605 5.60014328 -1.23081757  
[7] -2.01569423 1.60331611 3.20682778 -1.46176645
```

15.4.2. Mengatur tampilan dari output

Ada beberapa fungsi *built-in* yang dapat digunakan untuk mengatur tampilan output, baik dengan menampilkan layar maupun dengan menyimpan data di *disk*. Berikut adalah uraian tentang beberapa fungsi tersebut.

a. Fungsi **tab** dan **newline**

Untuk menggunakan tab dalam menampilkan output maka gunakan `"\t"` sedangkan untuk mengganti baris, gunakan `"\n"`. Perintah ini sering digunakan bersama dengan perintah **cat** (lihat bagian d).

b. Perintah **print**

Perintah ini bertujuan untuk menampilkan suatu objek ke layar sesuai dengan jenis data. Perhatikan contoh berikut ini.

```
> dataprint=list(karakter=letters[1:5], numerik=c(1:5))  
> print(dataprint)  
$karakter  
[1] "a" "b" "c" "d" "e"  
  
$numerik  
[1] 1 2 3 4 5
```

c. Perintah **format**

Perintah **format** bertujuan untuk mengubah *mode* data dari numerik ke karakter. Lihat contoh berikut ini.

```

> angka= 1:10
> angka
[1] 1 2 3 4 5 6 7 8 9 10
> karakter =format (angka)
> angka_karakter =format (angka)
> angka_karakter
[1] " 1" " 2" " 3" " 4" " 5" " 6" " 7" " 8" " 9" "10"

```

d. Perintah **cat**

Perintah **cat** merupakan perintah yang cukup fleksibel didalam menampilkan output di layar, yaitu dapat menampilkan data *character*, data numerik, komentar-komentar output, ataupun menuliskan data layar. Lihat *help* menu dari **R** untuk keterangan lebih lanjut mengenai fungsi ini. Untuk mengilustrasikan penggunaan dari perintah **cat** ini diberikan contoh sebagai berikut. Misalkan *user* ingin merubah tampilan dari output fungsi bangkitan normal diatas, dan akan dilakukan modifikasi dari fungsi itu seperti berikut ini.

```

datanormal3 <- function(n=10,mean=15,variansi=9)
{
  data<-rnorm(n,mean,sqrt(variansi))
  cat("=====\n")
  cat("  List Data \n")
  cat("=====\n")
  cat(data, sep = "\n")
  cat("=====\n")
}

```

Setelah file *disourcekan* (jika diketik melalui **R-Editor**), maka contoh output dari fungsi ini dapat dilihat dibawah ini. Perhatikan perbedaan tampilan output dengan perintah **datanormal3**. Perintah "**\n**" dan "**\t**" telah dikenalkan pada bagian (a) diatas.

```

> datanormal3()
=====
List Data
=====
2.422166
4.422396
.....
8.015969
=====

```

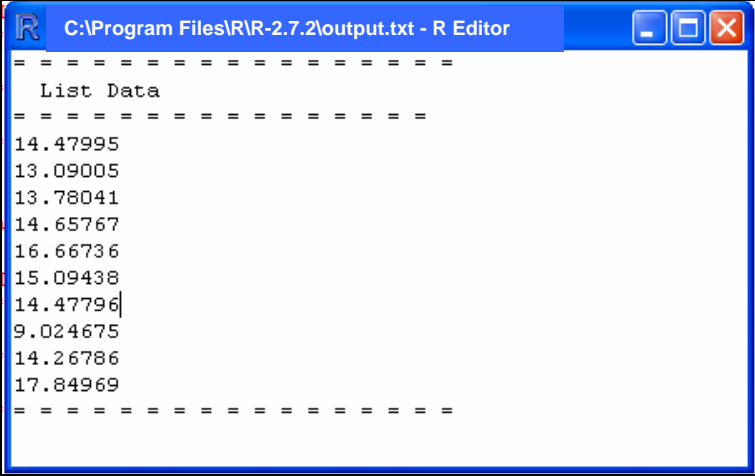
Lebih lanjut, fungsi **cat** dapat digunakan untuk menyimpan output kedalam suatu file eksternal *disk*. Misalkan output dengan format yang sama dengan contoh diatas ingin disimpan ke direktori **c:** dengan nama file **output.txt** (nama maksimum 8 *character*). Maka perlu dilakukan modifikasi fungsi **datanormal** menjadi sebagai berikut.

```
datanormal4 <- function(n=10,mean=15,variansi=9)
{
  data <- rnorm(n,mean,sqrt(variansi))
  cat("=====\n",file="c:output.txt")
  cat("  List Data \n", append=T,file="c:output.txt")
  cat("=====\n", append=T,
      file="c:output.txt")
  cat(data, sep = "\n", append=T,file="c:output.txt")
  cat("=====\n", append=T
      ,file="c:output.txt")
}
```

Perhatikan perbedaan fungsi **datanormal4** dan **datanormal3** pada contoh diatas. Lokasi dan nama file dapat disesuaikan dengan keinginan *user*. Setelah disourcekan file *script* dari fungsi ini, maka fungsi ini dapat dipanggil menggunakan perintah sebagai berikut.

```
> datanormal4()
```

Output dari fungsi **datanormal4** dapat diakses dengan melihat file **output.txt** melalui pilihan **File**, dan kemudian klik **Open script...** Perhatikan *directory* yang aktif ada di **C:\Program Files\R\R-2.7.2**



```
R C:\Program Files\R\R-2.7.2\output.txt - R Editor
=====
  List Data
=====
14.47995
13.09005
13.78041
14.65767
16.66736
15.09438
14.47796
9.024675
14.26786
17.84969
=====\\n
```


e. Perintah `write.table`

Fungsi ini dapat digunakan untuk menuliskan data (biasanya bertipe **dataframe**) yang ada kedalam suatu file di *disk*. Untuk keterangan lebih lanjut dari fungsi ini, lihat fasilitas *help* dari **R**. Berikut adalah contoh penggunaan dari perintah tersebut.

```
datanormal5 <- function(n=10,mean=15,variansi=9)
{
  data <- as.data.frame(rnorm)(n,mean,sqrt(variansi))
  write.table(data,file="c:output.txt",sep="\t")
}
```

Setelah disourcekan file *script* dari fungsi ini, maka selanjutnya fungsi ini dapat dipanggil dengan menggunakan perintah seperti berikut ini.

```
> datanormal5()
```

15.5. Contoh-contoh fungsi

Selanjutnya akan diberikan beberapa contoh fungsi dengan menggunakan perintah-perintah yang sudah dijelaskan di bagian sebelumnya. Contoh-contoh ini diharapkan dapat lebih memberi pemahaman tentang bagaimana mendefinisikan fungsi baru di **R**. Sebelum mengetikkan *script* fungsi ini pada **R-Editor**, perintah-perintah berikut disarankan untuk dicoba terlebih dahulu baris per baris pada **R-Console**. Jika urutan sudah benar, baik secara *logic* maupun *syntax*, maka urutan-urutan perintah pada **R-Console** dapat dicopy, kemudian dipaste ke **R-Editor**. Cara ini akan membantu proses mencari kesalahan dalam program (*debugging*).

▪ Uji t dengan sampel tunggal

Berikut ini adalah contoh fungsi untuk uji rata-rata sampel tunggal dengan varians tidak diketahui dengan menggunakan statistik uji **t**. Pada fungsi berikut, diberi nama **uji_t**, dengan argumen **x** sebagai vektor data yang akan diuji, **mu0** sebagai nilai rata-rata yang dihipotesiskan, dan **arah** sebagai identifikasi apakah hipotesis alternatif pengujian bersifat dua arah, kurang dari atau lebih dari.

```
uji_t = function(x,mu0,arah)
{
  df=length(x)-1
  T=abs((mean(x)-mu0)/(sd(x)/sqrt(length(x))))
  if (arah==0) P=2*(1-pt(abs(T),df))
  else if (arah==-1) P=pt(T,df)
  else P=pt(T,df,lower.tail=FALSE)
```

```

cat("Uji t Sampel Tunggal","\n")
cat("Ho:mu =",mu0,"\n")
if (arah==0) cat("H1:mu !=",mu0,"\n")
if (arah==-1) cat("H1:mu <",mu0,"\n")
if (arah==1) cat("H1:mu >",mu0,"\n")
cat("Mean=",mean(x)," , stdev=", sd(x),"
    n=",length(x),"\n")
cat("T = ",T," ,df = ",df," ,P = ",P,"\n")
}

```

Sebelum menjalankan, simpan terlebih dulu fungsi itu dengan nama **uji_t.R** dan kemudian *disourcekan*. Untuk menjalankan akan digunakan data bakteri seperti yang dibahas pada Bab 7. Sehingga, perbandingan hasil fungsi ini dapat dilakukan dengan output uji sampel tunggal pada R di Bab 7 tersebut.

```

> bakteri=c(175,190,215,198,184,207,210,193,196,180)
> uji_t(bakteri,200,0)

Uji t Sampel Tunggal
Ho:mu= 200
H1:mu!= 200
Mean = 194.8 stdev = 13.13858 n = 10
T = 1.251570 ,df = 9 P = 0.2422777

> uji_t(bakteri,200,1)

Uji t Sampel Tunggal
Ho:mu= 200
H1:mu> 200
Mean = 194.8 stdev = 13.13858 n = 10
T = -1.251570 ,df = 9 P = 0.8788612

> uji_t(bakteri,200,-1)

Uji t Sampel Tunggal
Ho:mu= 200
H1:mu< 200
Mean = 194.8 stdev = 13.13858 n = 10
T = -1.251570 ,df = 9 P = 0.1211388

```

▪ Regresi Linear Sederhana

Berikut ini akan diberikan contoh fungsi regresi untuk mengestimasi persamaan regresi linear sederhana. Fungsi ini mempunyai 2 argumen, yaitu **y** sebagai variabel respon dan **x** sebagai variabel prediktor. Estimasi dilakukan dengan metode kuadrat terkecil. Selanjutnya berturut-turut dihitung *standard error* hasil estimasi, **t** hitung dan nilai **p**. Sebagai validasi, hasil perhitungan dapat dibandingkan dengan output menu regresi seperti pada Bab 8.

```
regresi<-function(y,x)
{
  if (length(y)!=length(x)) cat("Banyak data tidak sama\n")
  k=1
  n=length(x)
  for (i in 1:n) k[i]=1
  X=cbind(k,x)
  b=(solve(t(X)%*%X))%*%t(X)%*%y
  y_hat=X%*%b
  e=y-y_hat
  SSR=sum((y_hat-mean(y))^2)
  SSE=sum((y-y_hat)^2)
  MSR=SSR/1
  MSE=SSE/(n-1-1)
  cov_b=solve(t(X)%*%X)*MSE
  se_b=sqrt(diag(cov_b))
  t_value=(1/se_b)*b
  p_value=2*(1-pt(abs(t_value),n-2))
  cat("          estimate std.error t_value    p    \n")
  for (i in 1:2)
  {
    if (i==1) cat("intercept
      ",b[i],se_b[i],t_value[i],p_value[i],"\n")
    else cat("      x
      ",b[i],se_b[i],t_value[i],p_value[i],"\n")
  }
}
```

Untuk menjalankan, misalkan digunakan data berikut ini.

```
> y=c(11,13,15,12,14,16,18)
> x=c(2,4,5,3,6,7,8)
> regresi(y,x)
```

	estimate	std.error	t_value	p
intercept	8.785714	0.7498299	11.71694	7.960169e-05
x	1.071429	0.1392399	7.694838	0.0005912413

DAFTAR PUSTAKA

- Cryer, J.D. (1986). *Time Series Analysis*. Boston: PWS-KENT Publishing Company.
- Draper, N.R. and Smith, H. (1981). *Applied Regression Analysis*. Second Edition, John Wiley & Sons, Inc.
- Granger, C.W.J. and Terasvirta, T. (1993). *Modeling Nonlinear Economic Relationships*. Oxford: Oxford University Press.
- Gujarati, D.N. (1996). *Basic Econometrics*. 5th edition, McGraw Hill International, New York.
- Hair, J.F., Anderson, R.E., Tatham, R.L. and Black, W.C. (2006). *Multivariate Data Analysis*. 6th edition, Prentice Hall International: United Kingdom.
- Hanke, J.E. and Reitsch, A.G. (2001). *Business Forecasting*. 7th edition, Prentice Hall, Englewood Cliffs, N.J.
- Härdle, W. (1991). *Smoothing Techniques with Implementation in S*. New York: Springer-Verlag.
- Hosmer, D.W. and Lemeshow, S. (1989). *Applied Logistic Regression*. New-York: John Wiley & Sons.
- Johnson, R.A. and Bhattacharyya, G.K. (1996). *Statistics: Principles and Methods*. 3rd edition, Canada: John Wiley & Sons.
- Johnson, N. and Wichern, D. (1998). *Applied Multivariate Statistical Analysis*. Prentice-Hall, Englewood Cliffs, N.J.
- Kutner, M.H., Nachtsheim, C.J. and Neter, J. (2004). *Applied Linear Regression Models*. McGraw Hill International, New York.
- Lee, T.-H., White, H., and Granger, C.W.J. (1993). Testing for Neglected Nonlinearity in Time Series Models: A comparison of Neural Network methods and alternative test. *Journal of Econometrics*, 56, pp. 269-290.
- Makridakis, S., Wheelwright, S. C. and Hyndman, R. J. (1998). *Forecasting: Method and Applications*. New York: John Wiley & Sons.
- McCullagh, P. and Nelder, J.A. (1989). *Generalized Linear Models*. Second edition. London: Chapman and Hall.
- Ramsey, J.B. (1969). Tests for Specification Error in Classical Linear Least Squares Regression Analysis. *Journal of the Royal Statistical Society, Series B*, **31**, 350–371.
- Scott, D.W. (1992). *Multivariate Density Estimation. Theory, Practice, and Visualization*. New York: John Wiley and Sons.
- Seber, G.A.F. and Wild, C.J. (1989). *Nonlinear Regression*. New York: John Wiley and Sons.

- Sharma, S. (1996). *Applied Multivariate Techniques*. New-York: John Wiley & Sons.
- Shumway, R.H. and Stoffer, D.S. (2006). *Time Series Analysis and Its Applications with R Examples*. Second edition, Springer: New York, USA.
- Silverman, B.W. (1985). *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.
- Terasvirta, T., Lin, C.-F., and Granger, C.W.J. (1993). Power of the neural network linearity test. *Journal of Time Series Analysis*, **14**, 159–171.
- Wand, M.P. and Jones, M.C. (1995). *Kernel Smoothing*. London: Chapman and Hall.
- Wei, W.W.S. (2006). *Time Series Analysis: Univariate and Multivariate Methods*. Second edition, Addison-Wesley Publishing Co., USA.
- White, H. (1989). An additional hidden unit test for neglected nonlinearity in multilayer feedforward networks. In *Proceedings of The International Joint Conference on Neural Networks*, Washington, DC (pp. 451-455). San Diego, CA: SOS Printing.

*** Berikut adalah referensi e-book yang dapat didownload di server CRAN R-Project**

- Baron, J. and Li, Y. (2003). *Notes on the use of R for psychology experiments and questionnaires*. Department of Psychology, University of Pennsylvania.
- Bliese, P. (2006). *Multilevel Modeling in R (2.2): A Brief Introduction to R, the multilevel package and the nlme package*. Paul.bliese@us.army.mil.
- Chongsuvivatwong, V. (2006). *Analysis of Epidemiological Data Using R and Epicalc*. Epidemiology Unit, Prince of Songkla University, Thailand.
- Faraway, J. J. (2002). *Practical Regression and Anova using R*. www.stat.lsa.umich.edu/~faraway/book.
- Farnsworth, G.V. (2006). *Econometrics in R*. g-farnsworth@kellogg.northwestern.edu.
- Maindonald, J.H. (2004). *Using R for Data Analysis and Graphics: Introduction, Code and Commentary*. Centre for Bioinformation Science, Australian National University.
- Owen, W.J. (2007). *The R Guide*. Department of Mathematics and Computer Science, University of Richmond.
- Paradis, E. (2005). *R for Beginners*. Institut des Sciences de l'Evolution, Universite Montpellier II, France.
- Ricci, V. (2005). *Fitting Distributions with R*. vito_ricci@yahoo.com.
- Rossiter, D.G. (2007). *Introduction to the R Project for Statistical Computing for use at ITS*. <http://www.its.nl/personal/rossiter>.
- Seefeld K. and Linder, E. (2007). *Statistics Using R with Biological Examples*. Department of Mathematics & Statistics, University of New Hampshire, Durham, NH.

Venables, W.N. and Smith, D.M. (2007). *An Introduction to R*. The R Development Core Team.

Verzani, JA. (2002). *Simple R - Using R for Introductory Statistics*. www.math.csi.cuny.edu/Statistics/R/simpleR/Simple.

Vikneswaran (2005). *An R companion to "Experimental Design"*. www.geocities.com/vik

DAFTAR INDEKS

- abline, 166, 179
- acf, 199-202
- AIC, AICc, 225
- Akaike's Information Criterion, *lihat* AIC
- Analisis,
 - Cluster, 234-236
 - Diskriminan, 232-233
 - Faktor, 230-231
 - Multivariat, 230
 - Regresi, 128-157
 - Runtun Waktu, 184-229
 - Variansi, 110-117
- ANOVA, *lihat* Analisis Variansi
- ARIMA, 198-226
 - cek diagnosa, 207-209
 - estimasi, 205
 - identifikasi, 199-202
 - musiman, 216-224
 - nonmusiman, 203-216
 - peramalan, 210-211
 - Yule-Walker, 205
 - least squares, 205-206
 - maksimum likelihood, 205-206
- Bartlett, 120-121
- bar-chart, 56-57
- binomial, 70-78
- box-and-whisker plot, 50-51
- Box-Jenkins methodology, 198
- chi-square test, 94-96
- compute, 27-29
- correlation, 88-90
- data, 30-37
 - array, 30
 - frame, 34
 - matriks, 31
 - list, 37
- dataset, 24-25
- dendogram, 235
- diagram,
 - batang (bar-chart), 56-57
 - batang dan daun, 48-49
 - lingkaran (pie-chart), 57-58
 - pencar (scatter-plot), 53-54
- direktori, 6
- diskriminan, 232-234
 - linear, 233-234
- distribusi, 61-79
 - binomial, 70-78
 - diskrit, 70
 - frekuensi, 85-86
 - kontinu, 62
 - normal, 62-69
- edit,
 - data, 20
- eksponensial smoothing, 187-197
 - ganda, 195-196
 - sederhana, 196-197
 - Holt-Winters, 189-194
- entry,
 - data, 16-18
- estimasi,
 - densitas, 237-241
 - model ARIMA, 205
 - model linear tergeneralisir, 161-165
 - model non-linear, 256-259
 - regresi linear, 128-136
- Factor Analysis, 230-231
- factors, 42
- frame, 34
- fungsi,
 - distribusi, 61,62,69
 - plot, 168-180
- Generalized Linear Model, 158-165
- grafik, 43-60, 168-183
 - setting, 182-183
- GLM, 158-165

- help, 10-15
 - search-engine, 12-14
 - online search-engine, 15
- histogram, 46
- Holt-Winters, 189-194
 - aditif, 189-192
 - multiplikatif, 193-194
- import data,
 - ASCII, 38
 - EXCEL, 20,39
 - MINITAB, 22, 41
 - SPSS, 22,41
- instalasi R, 2
- kenormalan,
 - shapiro-wilk, 91
- kernel, 237-240
 - Epanechnikov, 238-239
 - Gaussian, 238-239
- kontingensi, 92-96
- kuantil, 62,71
 - binomial, 71
 - normal, 62
- lda, 233
- Levene, 121
- library, 1
- linear,
 - model, 132-137
 - regresi, 128-131
- lm, 132-137
- logistik,
 - regresi, 161-165
- matriks, 31,86
 - data, 31
 - korelasi, 88
- mean,
 - plot, 55
- model, 128-165, 185-226, 256-276
 - ARIMA, 198-226
 - eksponensial smoothing, 187-197
 - linear, 132-137
 - linear tergeneralisir, 158-165
 - regresi linear, 128-131
 - regresi non-linear, 256-276
 - tren linear, 185-186
- multivariat,
 - analisis, 230-236
- non-linear, 256-276
 - model, 256-262
 - uji, 263-276
- normal, 62-69, 91
 - distribusi, 62-69
- nls, 259-262
- operator, 279
- pacf, 199-202
- paket, 1
- pemrograman, 277-291
 - fungsi, 277-279
 - indeks, 58
 - optional argument, 284-285
 - required argument, 285-286
- plot, 55, 58, 168-180
 - indeks, 58
 - interaktif, 80
 - rata-rata, 55
 - utama, 168-178
 - tambahan, 179
- proporsi, 122-127
 - uji perbedaan, 122-127
- qda, 233
- QQ-plot, 51
- Ramsey's, 263-266
 - RESET, 263-266
- recode, 25
- regresi, 128-157, 237-248, 256-262
 - kernel, 241-243
 - linear, 128-157
 - non-linear, 256-262
 - nonparametrik, 241-248
 - spline, 243-248
- ringkasan, 81-84
 - numerik, 81-84

runtun waktu, 184-229

Shapiro-Wilk, 88

SIC, 203

spline, 221-233

 basis, 227-228

 B-spline, 223-225

 kubik, 226-229

SSasympOrig, 239-240

Stem-and-Leaf, 48

summary, 79-81

tabel, 84,89

 kontingensi, 89

 statistika, 84

Terasvirta, 274-276

transformasi, 25

 dataset, 25

uji non-linearitas, 263-276

 Ramsey's RESET, 263-266

 Terasvirta, 274-276

 White, 267-273

uji proporsi, 122-127

 dua sampel, 125-127

 sampel tunggal, 122-124

uji rata-rata, 99-117

 dua sampel bebas, 102-106

 sampel berpasangan, 107-109

 sampel tunggal, 99-101

 One-way ANOVA, 110-114

 Multi-way ANOVA, 115-117

uji variansi, 112-115

 Bartlett, 114

 dua variansi, 112

 Levene, 115

vektor, 30

White, 266-272

Yule-Walker, 205

TENTANG PENULIS



Suhartono, bekerja sebagai dosen di Jurusan Statistika, Institut Teknologi Sepuluh Nopember (ITS). Ia lulus S1 Statistika ITS pada tahun 1995 dan mendapat *Master of Statistical Analysis and Stochastic Systems* dari *University of Manchester Institute of Science and Technology* (UMIST), UK, tahun 1998. Kemudian ia mendapat Dr pada tahun 2007 di bidang *Neural Network for Time Series Forecasting* dari Jurusan Matematika, Universitas Gadjah Mada (UGM).

Bidang penelitian yang banyak dilakukan adalah *time series forecasting*, *neural network for data analysis*, *spatial time series*, dan *econometrics time series*. Beberapa area terapan yang menjadi obyek penelitian dan telah dilakukan antara lain pemodelan inflasi di Indonesia, turism di Bali, transportasi (kendaraan) di jalan tol, hidrologi (debit air di suatu bendungan), dan pemodelan pemakaian energi listrik jangka pendek di suatu area distribusi. Saat ini ia sedang meneliti tentang model *hybrid neural network* dan analisis *wavelet* untuk *time series forecasting*, serta mengembangkan model intervensi, *structural change*, dan model variasi kalender yang banyak terjadi di beberapa kasus data series di Indonesia.

Di Jurusan Statistika ITS, ia saat ini menjadi Kepala Laboratorium Statistik Komputasi. Beberapa mata kuliah yang diampu adalah Analisis Runtun Waktu (*Time Series Analysis*), Analisis Data, dan Analisis Multivariat di program Sarjana dan Magister. Beberapa file untuk pembelajaran *Time Series Analysis* dan *Design of Experiment* yang telah ia buat dapat diakses secara online pada open content ITS melalui www.its.ac.id. Selain itu, karya-karya penelitian yang telah ia lakukan dapat juga diakses secara online pada pilihan *web personal* dosen di *homepage* ITS tersebut.