

Dasar-dasar Pemrograman *Datamining* di R: Interface Pemrograman Java

Sigit Wahyu Kartiko

gsigit[at]gmail[dot]com

Lisensi Dokumen:

Copyright © 2003-2006 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Pendahuluan

Selain sebagai software paket statistika, R juga merupakan lingkungan sistem pemrograman yang cukup lengkap. Artinya, R dapat digunakan sebagai alat untuk memecahkan berbagai masalah melalui pemrograman. Tentu saja masalah yang lebih tepat dipecahkan dengan R adalah yang terkait dengan analisis data dalam konteks statistika.

Para developer R menyadari bahwa analisis data kuantitatif melalui statistika memerlukan beberapa tahapan yang mana setiap tahapan dibutuhkan alat bantu untuk 'berbicara' dengan lingkungan disekitarnya. Secara singkat tahapan ini meliputi 1) persiapan data, 2) pemilihan metode/teknik, 3) eksekusi dan 4) penyajian informasi. Sebagai contoh, pada tahap persiapan data, alat bantu yang dibutuhkan bagi seorang stastisi adalah kemudahan dalam menghubungkan *raw* data (database, file terstruktur, dsb) dengan lingkungan R. Solusi yang diberikan oleh developer dan kontributor R terkait dengan hal ini diantaranya adalah membuat library ODBC, JDBC, CSV dll.

Selain kemudahan, konsep yang sering digunakan oleh para programer adalah otomatisasi dan integrasi proses. Artinya dalam sistem yang kompleks, lingkungan R harus bisa menyediakan interface bahasa yang berfungsi sebagai penghubung antar fungsi-fungsi dalam sistem tersebut. Dengan latar belakang itulah developer dan kontributor R membuat berbagai interface bahasa pemrograman yang berada diluar R seperti C, Java dan Tcl/Tk agar bisa memudahkan proses otomatisasi dan integrasi.

Pada kesempatan kali ini penulis ingin mengkhususkan pembahasan interface bahasa di R yang sangat populer yaitu java. Diharapkan para pembaca sudah menyempatkan diri untuk mengulas materi sebelumnya yang sudah diupload oleh penulis diantaranya: pengenalan R¹, dasar-dasar

1 <http://ilmukomputer.org/2012/04/06/mengenal-software-statistika-r-sebagai-datamining-tooldi-linux/>

pengelolaan data² dan plotting³.

Instalasi Java Development Kit (JDK)

Sebelum melakukan instalasi JDK di linux, langkah pertama yang harus dikerjakan adalah mendownload terlebih dahulu file JDK tersebut di situs Oracle. Selanjutnya langkah-langkah instalasi dapat ditelusuri di bagian “Self-Extracting Binary Installation for Linux”⁴ yang juga berada di situs Oracle.

Instalasi Plugin-plugin Terkait

Setelah java terpasang dengan benar, langkah berikutnya adalah menginstalasi plugin-plugin yang terkait dengan interface pemrograman java di R. Plugin tersebut antara lain:

Plugin	Deskripsi
DBI	Low-level database interface (DBI) yang mampu menghubungkan antara R dengan RDBMS
rJava	Low-level interface untuk java
RJDBC	Koneksi java dengan database (JDBC). Selain itu perlu ditambahkan driver JDBC untuk database yang relevan dalam menggunakan plugin ini
Rserve	Merupakan server R secara lokal yang memanfaatkan komunikasi TCP/IP sehingga memungkinkan bagi program di client untuk melakukan perintah ke server R tanpa membuka workspace R. Rserve ini memiliki kelebihan bahwa akses terhadap server bersifat thread safe. Namun hanya direkomendasikan untuk R yang berjalan di linux. Implementasi di sisi client yang tersedia adalah C/C++ dan Java.

Tabel 1: Plugin minimal yang dibutuhkan dalam pemrograman interface java di R

Sebelum melakukan instalasi pastikan konfigurasi java di R sudah dibuat. Ketikkan perintah berikut ini di terminal linux untuk mengkonfigurasi environment java:

```
R CMD javareconf
```

Berikutnya adalah menginstalasi file-file plugin. Ada baiknya jika menggunakan user root untuk memudahkan proses kompilasi file-file plugin. Untuk hal ini maka perlu di-setting R_HOME untuk user root dengan melakukan *export* variabel R_HOME dalam file /root/.bashrc⁵.

Perintah instalasi file-file plugin dapat dilakukan dengan sekali perintah yaitu sebagai berikut:

1. Berada di direktori tempat menaruh file-file plugin

```
cd /home/sigit/R/src/plugin/
```

2. Perintah instal plugin

2 <http://ilmukomputer.org/2012/04/08/dasar-dasar-pemrograman-datamining-di-r-pengelolaan-data/>

3 <http://ilmukomputer.org/2012/04/13/dasar-dasar-pemrograman-datamining-di-r-plotting/>

4 <http://www.oracle.com/technetwork/java/javase/install-linux-self-extracting-138783.html>

5 Hal ini sudah dibahas di tulisan sebelumnya mengenai pengenalan software statistika R di <http://ilmukomputer.org/2012/04/06/mengenal-software-statistika-r-sebagai-datamining-tooldi-linux/>

```
R CMD INSTALL -l /home/sigit/R/R-2.10.0/program/lib/R/library/ \
DBI_0.2-4.tar.gz rJava_0.8-1.tar.gz RJDBC_0.1-5.tar.gz Rserve_0.6-0.tar.gz
```

Langkah berikutnya yang sangat penting bagi client yang akan mengakses server R adalah membuat library java dari plugin Rserve. Library java berupa file jar dibuat dari source plugin Rserve. Lakukan langkah-langkah berikut untuk mengkompilasi source menjadi file jar.

1. Kembali berada di direktori tempat menaruh file-file plugin

```
cd /home/sigit/R/src/plugin/
```

2. Ekstrak isi file Rserve

```
tar -xzf Rserve_0.6-0.tar.gz
```

3. Berada di direktori source java di folder Rserve hasil ekstraksi

```
cd Rserve/src/client/java-new
```

4. Kompilasi file source dengan perintah make

```
make
```

Tunggu beberapa saat, setelah sukses maka hasil kompilasi tersebut adalah 2 file yang penting dalam melakukan pemrograman java di R yaitu file **REngine.jar** dan **Rserve.jar**.

Menghidupkan Server Lokal R

Supaya program di sisi client dapat berkomunikasi dengan layanan R melalui TCP/IP maka server R harus dihidupkan. Cara menghidupkan server lokal R dapat dilakukan dengan 2 cara yang pertama, masuk dalam workspace R terlebih dahulu lalu dihidupkan servernya. Atau cara kedua melalui perintah R di terminal tanpa masuk ke workspace langsung menghidupkan server.

Cara Pertama

Cara pertama adalah masuk terlebih dahulu ke dalam workspace R dengan perintah:

```
R
```

setelah berada di dalam workspace ketikkan panggil library Rserve lalu aktifkan dengan fungsi **Rserve()**:

```
> library(Rserve)
> Rserve()
```

Jika berhasil maka Rserve akan hidup dalam mode daemon menggunakan port 6311:

```
Starting Rserve on port 6311 :
/home/sigit/R/R-2.10.0/program/lib/R/bin/R CMD /home/sigit/R/R-
2.10.0/program/lib/R/library/Rserve/libs//Rserve-bin.so
[Previously saved workspace restored]
Rserv started in daemon mode.
```

Cara Kedua

Cara berikutnya adalah langsung menjalankan perintah R:

```
R CMD Rserve
```

Karena belum ada fasilitas untuk mematikan Rserve maka untuk menghentikan server gunakan perintah sebagai berikut:

```
killall Rserve
```

Operasi Tipe Data R melalui Java

Pengelolaan variabel/data di R dapat dilakukan melalui pemrograman Java. Tipe-tipe variabel/data tersebut antara lain adalah *scalar*; *vector* (numerik, karakter, logical), matriks, *dataframe/list* dan *factor/ordered*.

Untuk dapat melakukan operasi variabel/data maka object java yang harus di-*instance* adalah ***org.rosuda.REngine.Rserve.RConnection***. *RConnection* merupakan class yang memiliki metode untuk me-*assign* variabel, diantaranya adalah:

- `assign()`
- `eval()`
- `parseAndEval()`

Penggunaan masing-masing metode tersebut dapat dimengerti dalam contoh berikut ini.

Tipe Data Vector

Untuk membuat data vector di R dengan di java akan ditampilkan dalam tabel yang dipersandingkan sebagai berikut:

R	Java
<pre>kolom.1 <- c(1,2,3) kolom.2 <- c(1.0, 2.0, 3.0) vector.a <- c(kolom.1, kolom.2)</pre>	<pre>RConnection c = new RConnection(); RList l = new RList(); l.put("kolom.1", new REXPInteger(new int[]{ 1,2,3})); l.put("kolom.2", new REXPDouble(new double[]{ 1.0,2.0,3.0})); c.assign("vector.a", new REXPGenericVector(l));</pre>

Tabel 2: assign data vector di R dan Java

Output dari assign variabel tersebut adalah berupa vector dengan tipe data integer, sedangkan yang bertipe double akan menjadi integer karena yang pertama kali di masukkan ke dalam vector di variabel `vector.a` adalah variabel `kolom.1` yang bertipe integer. Outputnya adalah sebagai berikut:

```
> [1] 1 2 3 1 2 3
```

Tipe Data Frame

Hampir sama dengan data vector, untuk membuat data frame dibutuhkan data perkolom terlebih dahulu melalui *RList*, lalu di *assign* ke dalam *RConnection* dengan menggunakan variabel data frame yaitu `REXP.createDataFrame()`.

R	Java
<pre>kolom.1 <- c(1,2,3) kolom.2 <- c(1.0, 2.0, 3.0) data.frame.a <- c(kolom.1, kolom.2)</pre>	<pre>RConnection c = new RConnection(); RList l = new RList(); l.put("kolom.1", new REXPInteger(new int[]{ 1,2,3})); l.put("kolom.2", new REXPInteger(new double[]{ 1.0,2.0,3.0})); c.assign("data.frame.a", new REXP.createDataFrame(l));</pre>

Tabel 3: assign data frame di R dan Java

Output data frame tersebut adalah sebagai berikut:

	<i>kolom.1</i>	<i>kolom.2</i>
1	1	1
2	2	2
3	3	3

Cara lain untuk me-assign variabel vector adalah dengan memanfaatkan metode `parseAndEval()`. Cara ini sangat praktis apabila sebab seolah-olah kita bekerja pada workspace R. Semua perintah R dapat dilakukan didalam metode dengan menambahkan karakter kurung siku buka (`{`) dan tutup (`}`), pemisah antar baris-nya adalah semicolon(`;`), dan yang ditangkap sebagai variabel REXP di java adalah baris terakhir. Sebagai misal data frame 'iris' adalah bawaan dari workspace R, maka dapat langsung dipanggil dengan metode `parseAndEval()`.

Java
<pre>RConnection c = new RConnection(); REXP df = c.parseAndEval("{data(iris); iris}"); c.assign("data.frame.iris", df);</pre>

Tabel 4: assign data frame dengan parseAndEval()

Dapat dilihat bahwa statement dalam `parseAndEval` adalah `"data(iris); iris"` yang diapit oleh tanda `{` dan `}`. Baris statement tersebut ada 2 yaitu `data(iris)` dan `iris`. Di workspace R, setelah mengimpor data iris dengan perintah `"data(iris)"` maka perintah berikutnya yaitu `"iris"` akan menghasilkan output tampilan data frame di layar. Tampilan dari perintah di baris terakhir inilah yang ditangkap oleh REXP sebagai variabel.

Plotting R melalui Java

Langkah-langkah melakukan plotting R di java dapat disederhanakan sebagai berikut:

1. Menentukan nama, ukuran, dan tipe file image plot yang akan dibuat
2. Memanggil variabel-variabel data frame yang akan ditampilkan dalam plot
3. Menyimpan stream file image ke dalam variabel di program java
4. Menampilkan stream file image.

Untuk lebih jelasnya akan kita sandingkan perintah di R dengan java sebagai berikut:

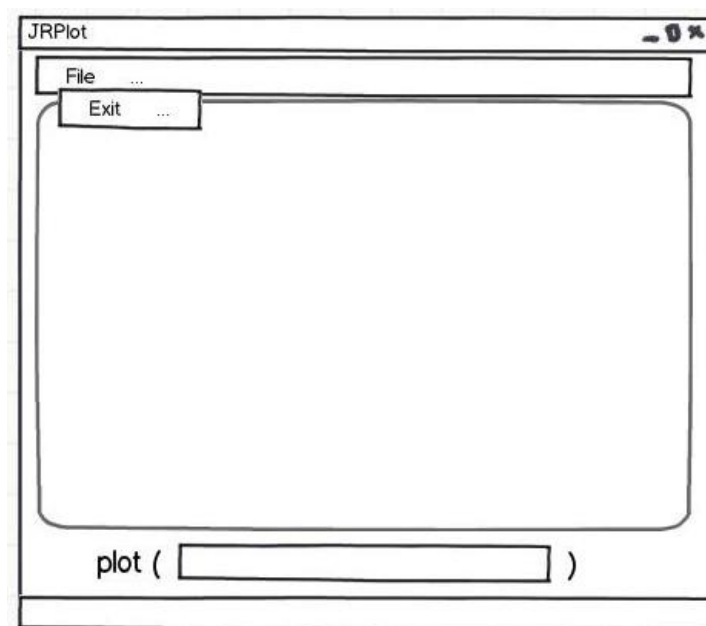
Langkah	R	Java
Menentukan nama, kualitas, tipe file image plot yang akan dibuat	<code>jpeg('test.jpg', quality=90)</code>	<pre>RConnection c = new RConnection(); REXP rImage =c.parseAndEval("jpeg('test.jpg','+ quality=90)");</pre>
Memanggil variabel-variabel data frame yang akan ditampilkan dalam plot	<code>data(iris) attach(iris) plot(Sepal.Length, Petal.Length, col=unclass(Species)) dev.off()</code>	<pre>c.parseAndEval("data(iris); attach(iris);"+ "plot(Sepal.Length, Petal.Length,"+ "col=unclass(Species)); dev.off()");</pre>
Menyimpan stream file image ke dalam variabel di program java		<pre>REXP rImage =c.parseAndEval("r=readBin('test.jpg','raw',1024*1024);"+ "unlink('test.jpg'); r")</pre>
Menampilkan stream file image.		<pre>Image img = Toolkit.getDefaultToolkit() .createImage(rImage.asBytes());</pre>

Tabel 5: assign data frame di R dan Java

Studi Kasus: Program Java Desktop menampilkan Grafik Fungsi di R

Agar lebih jelas, mari kita coba membuat program sederhana menampilkan grafik fungsi dengan java. Sebagai latihan program sederhana ini berbasis desktop dengan memanfaatkan *library* desktop swing di java.

Sebagai gambaran tampilan aplikasi ini adalah berupa form yang memiliki menu yang menampilkan item exit aplikasi, canvas sebagai tempat menampilkan output plot, dan textbox sebagai input fungsi yang diinginkan.



Gambar 1: sketsa program sederhana JRPlot

Source code selengkapnya adalah sebagai berikut:

File : JRPlot.java

```
//library desktop swing
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

//dari REngine.jar
import org.rosuda.REngine.*;
//dari Rserver.jar
import org.rosuda.REngine.Rserve.*;

public class JRPlot {
    JMenuBar menu;
    JMenu menuFile;
    JMenuItem menuItemKeluar;
    JTextField textRumus;
    JFrame window;
    JLabel labelPlot;
    RConnection c;

    public JRPlot() {
        // membuat menu standar
        menu = new JMenuBar();
        menuFile = new JMenu();
        menuItemKeluar = new JMenuItem();
        menuFile.setText("File");
        menuItemKeluar.setText("Tutup");
        menuFile.add(menuItemKeluar);
        menu.add(menuFile);

        // window
        window = new JFrame("Plot R ");
        window.setPreferredSize(new Dimension(400, 400));
        window.setJMenuBar(menu);

        // text box tempat menulis rumus
        textRumus = new JTextField("function(x){x^2}");
        textRumus.setFont(new Font("Arial", 20, 20));

        // tempat gambar plot
        labelPlot = new JLabel();
        window.getContentPane().add(labelPlot, BorderLayout.CENTER);
        window.getContentPane().add(textRumus, BorderLayout.SOUTH);

        //inisialisasi aksi, lihat method 'public void initAction()';
        initAction();
        window.pack();
        window.setLocationRelativeTo(null);
        window.setVisible(true);
    }

    //prosedur menggambar plot;
    private void gambarPlot(){
        try {
            //default koneksi ke localhost;
            c = new RConnection();
            int x = labelPlot.getSize().height;
            int y = labelPlot.getSize().width;

            // perintah script R yang paling penting
            REXP xp = c.parseAndEval("try(jpeg('test.jpg',,,h=" + x + ",w=" + y + "))");
            c.parseAndEval("plot(" + textRumus.getText() + ")");
            c.parseAndEval("dev.off()");
            xp = c.parseAndEval("r=readBin('test.jpg','raw',1024*1024); unlink('test.jpg'); r");

            //gambar ditangkap oleh java

```

```
Image img = Toolkit.getDefaultToolkit().createImage(xp.asBytes());
Icon plot = new ImageIcon(img);

//menaruh gambar di label tempat plot ditaruh
labelPlot.setIcon(plot);
window.setTitle("Plot R [" + textRumus.getText() + "]");
} catch (Exception e) {
JOptionPane.showMessageDialog(window, "maaf ada kesalahan!..");
e.printStackTrace();
}

}

private void initAction(){
//aksi menggambar plot jika textbox di tekan enter
textRumus.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
gambarPlot();
}
});

//aksi menggambar plot frame di ubah tinggi dan lebarnya
labelPlot.addComponentListener(new ComponentAdapter() {
public void componentResized(ComponentEvent e) {
gambarPlot();
}
});

window.addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent evt) {
tutupAplikasi();
}
});

menuItemKeluar.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
tutupAplikasi();
}
});

}

private void tutupAplikasi(){
//tutup koneksi dan aplikasi
c.close();
System.exit(0);
}

public static void main(String[] args){
//jalankan program
new JRPlot();
}
}
```

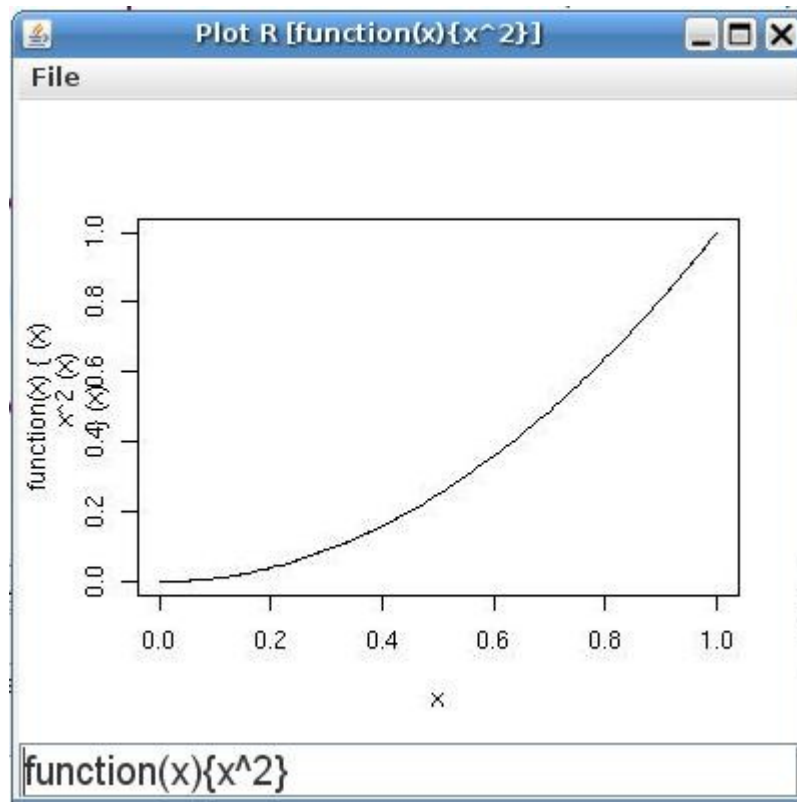
Sebelum melakukan kompilasi letakkan file Rserve.jar dan Rengine.jar sejajar dengan file JRPlot.java dan lakukan kompilasi program java dengan perintah sebagai berikut:

```
javac -cp Rserve.jar:REngine.jar:. JRPlot.java
```

File yang dihasilkan adalah berupa file-file bytecode dengan ekstensi .class. Hal ini berarti kompilasi telah berhasil dilakukan. Setelah itu jalankan program java dengan perintah sebagai berikut:

```
java -cp Rserver.jar:REngine.jar:. JRPlot
```

Tampilan program yang dihasilkan adalah sebagai berikut:



Gambar 2: output keluaran dari program JRPlot

Cobalah untuk mengetikkan fungsi lain misalnya “ $y = x^3$ ” maka gantilah dengan perintah sebagai berikut “ `function(x){x^2}` ” pada textbox fungsi lalu tekan enter. Maka grafik akan berubah.

Selain itu jika anda mengubah ukuran dari frame program maka tampilan grafik akan mengikuti lebar dan tinggi frame program tersebut

Penutup

Sampai dengan saat ini anda sudah dapat melakukan komunikasi antara bahasa R dan java. Teknologi interface java di R membantu developer untuk mengeksplor lebih jauh utility R di java sehingga integrasi antara kekuatan java dan R dapat digabungkan. Tidak mustahil apabila perintah-perintah R dijalankan secara otomatis melalui program java. Selain itu dengan memanfaatkan kemampuan java atas infrastruktur pemrograman web maka proses datamining di R melalui dunia maya sangat mungkin untuk dilakukan.

Referensi

1. Ihaka, R. & Gentleman, R. (1996). "R: A Language for Data Analysis and Graphics". *Journal of Computational and Graphical Statistics* 5 (3): 299–314. www.jstor.org

2. Rossiter, D. G. (2009). *Introduction to the R Project for Statistical Computing for use at ITC*. Accessed:03-03- 2010. [http:// www.itc.nl/ personal/ rossiter](http://www.itc.nl/personal/rossiter)
3. Williams, Graham J. *Rattle: A Data Mining GUI for R*. Accessed:06-04-2012. http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf
4. Chihara, Laura. *Plotting Concepts in R*. Accessed: 07-12-2009. <http://www.people.carleton.edu/~lchihara/Splus/RPlot.pdf>
5. <http://www.rforge.net/Rserve>
6. <http://www.R-project.org>

Biografi Penulis



Sigit Wahyu Kartiko. Menyelesaikan DIV di Sekolah Tinggi Akuntansi Negara dan Magister Ekonomi di Fakultas Ekonomi Universitas Indonesia. Memiliki kompetensi pada bidang *programming* (java, scala, R, C, C++), *database*, *accounting* (privat, publik), *public sector economics*.