

MODUL KOMPUTASI STATISTIKA LANJUT
GUI R: PEMBUATAN HISTOGRAM DAN PLOT



Disusun Oleh:
Ali Mahmudan
NIM. 24050118120027

DEPARTEMEN STATISTIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO
SEMARANG
2021

PENGANTAR RSHINY

R merupakan bahasa pemrograman statistika yang diciptakan oleh Ross Ihaka dan Robert Gentleman dari Departemen Statistika, Universitas Auckland, New Zealand yang dapat digunakan untuk analisis dan manipulasi data statistika (pemodelan statistika) serta grafik (Gio & Effendie, 2017). Salah satu program dalam R yang bisa membuat menu *User Interface* yang berbasis web adalah *R-shiny*. Menurut Tirta (2014), *R-shiny* mempunyai komponen yang dibagi menjadi dua kelompok, yaitu:

a. *User Interface* (UI)

Manfaat *User Interface* adalah sebagai berikut:

- 1) Panel kontrol adalah panel untuk mengontrol input berupa data, variabel, model, tergantung kompleksitas modul. Input data atau memasukkan data pada pengguna aplikasi *R-Shiny* dilakukan dengan perintah pada fungsi yang telah diberi identitas atau id (`input$id<-()` pada *server*).
- 2) Pemasukan permintaan nilai input (data dengan berbagai jenis variabel yang diperlukan, pemilihan model, jenis, dan kriteria uji statistika).
- 3) Penyajian output terkait hasil analisis atau uji. Setelah data terinput maka data inputan akan diproses sesuai kebutuhan dan hasil proses yang dilakukan akan didefinisikan menggunakan perintah `output$(id output)<-()` (Hendrian, 2021). Id *output* merupakan identitas yang digunakan untuk memanggil output hasil proses ke fungsi UI.

b. *Server*

Bagian ini merupakan otak dari program yang bertugas melakukan simulasi, berbagai analisis data sesuai pilihan pengguna dan selanjutnya mengirim hasilnya ke bagian output.

FUNGSI-FUNGSI DASAR PADA SHINY

✓ **User Interface**

`fluidPage`: untuk mengatur segala tampilan

`navbarPage`: membuat navbar

`tabpanel`: membuat tab

`sidebarLayout`: mengatur tata letak

sidebarPanel: membuat tata panel samping

fileInput: input berupa file

mainPanel: membuat panel utama

tabsetPanel: membuat anak tab

verbatimTextOutput: pembuatan output berupa tulisan

tableOutput: pembuatan output berupa tabel

plotOutput: pembuatan output berupa plot

✓ **Server**

output\$*id*: proses pembuatan output dengan memanggil *id* di UI

renderTable: pembuatan output berupa tabel

renderPrint: pembuatan output berupa tulisan

renderPlot: pembuatan output berupa plot

✓ **Running App**

shinyApp(ui,server)

CONTOH PENERAPAN

Persiapan Awal

```
install.packages("shiny")  
install.packages("shinythemes")
```

Library Paket-Paket yang Dibutuhkan

```
library(shiny)  
library(shinythemes)#JIKA INGIN MENGGUNAKAN TEMA
```

User Interface

```
tampilan<-fluidPage(  
  titlePanel("Membuat Histogram"),  
  sidebarLayout(  
    sidebarPanel(  
      sliderInput("jml","Pilih Jumlah Bilangan:",  
                  min = 1,max = 100,value = 50)),  
    mainPanel(  
      h4("Histogram"),plotOutput("hist")
```

```
)))
```

Server

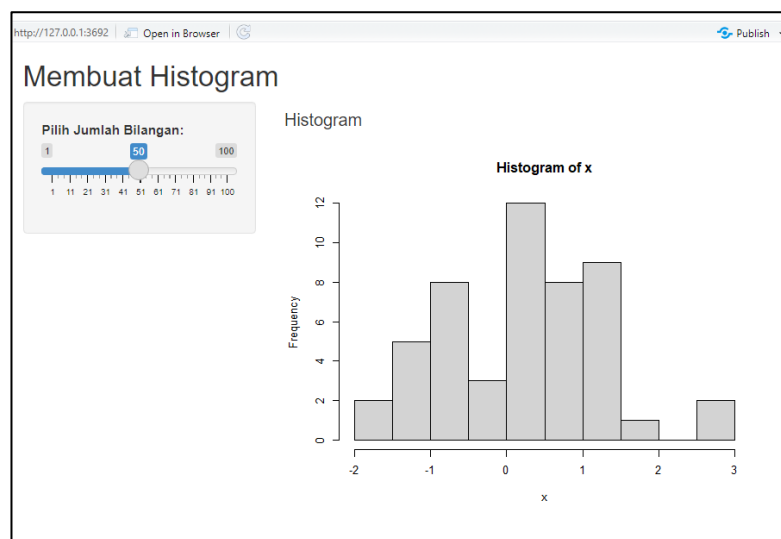
```
rancangan<-function(input,output){  
  output$hist<-renderPlot({  
    x=rnorm(input$jml)  
    hist(x)  
  })  
}
```

Running App

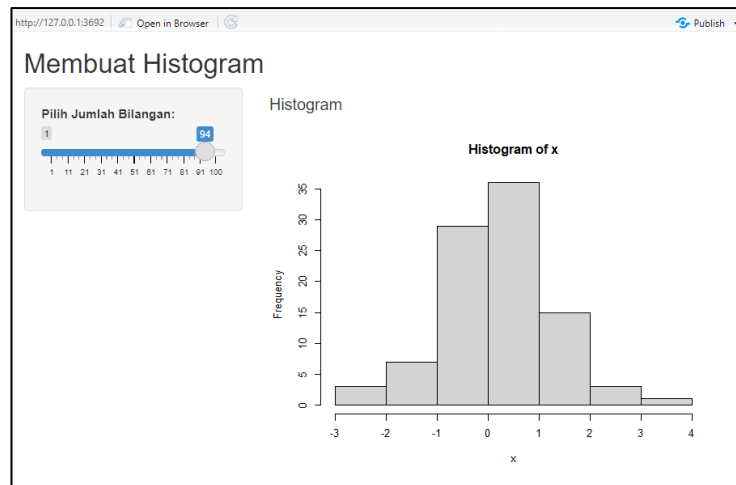
```
shinyApp(tampilan,rancangan)
```

TAMPILAN GUI R “Membuat Histogram”

Tampilan Awal



Tampilan Setelah Diatur Jumlah Bilangan (sesuai kebutuhan)



Membuat Histogram dengan Pilihan Warna

Persiapan Awal

```
install.packages("shiny")
install.packages("shinythemes")
```

Library Paket-Paket yang Dibutuhkan

```
library(shiny)
library(shinythemes)#JIKA INGIN MENGGUNAKAN TEMA
```

User Interface

```
tampilan<-fluidPage(
  titlePanel("Membuat Histogram 3 Pilihan Warna"),
  sidebarLayout(
    sidebarPanel(
      sliderInput("jml","Pilih Jumlah Bilangan:",
                  min = 1,max = 100,value = 50),
      selectInput("pilih",label="Pilih
Warna",choices=c("abu-abu","kuning","biru"))),
    mainPanel(
      h4("Histogram"),plotOutput("hist")
    )
  )
)
```

Server

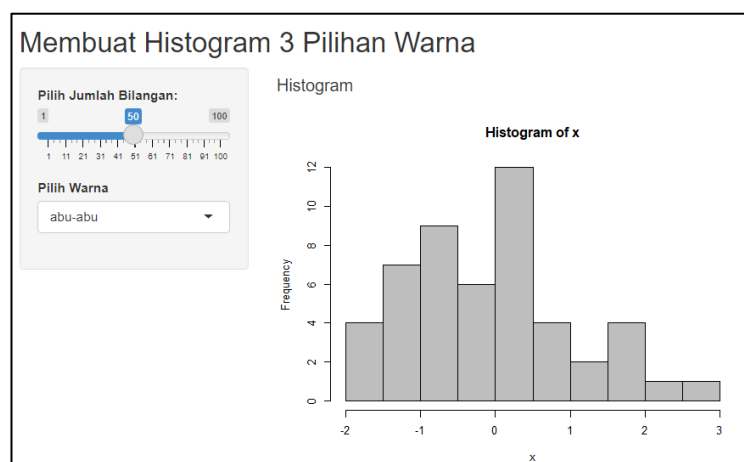
```
rancangan<-function(input,output){  
  output$hist<-renderPlot({  
    x=rnorm(input$jml)  
    warna=input$pilih  
    if(warna=="abu-abu"){  
      hist(x,col = "grey")  
    }  
    if(warna=="kuning"){  
      hist(x,col = "yellow")  
    }  
    if(warna=="biru"){  
      hist(x,col = "blue")  
    }  
  })  
}
```

Running App

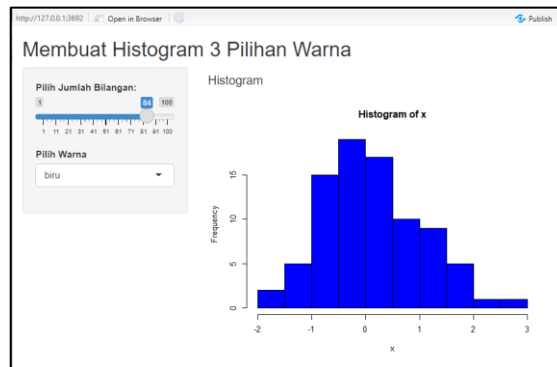
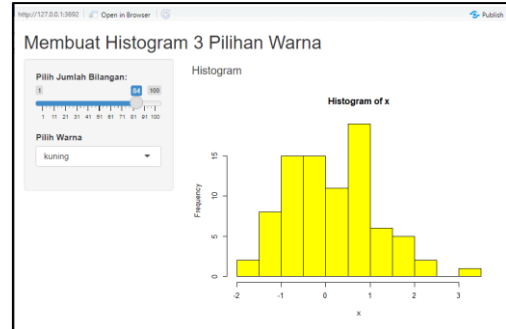
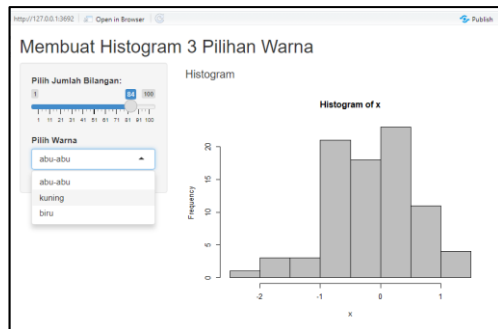
```
shinyApp(tampilan,rancangan)
```

TAMPILAN GUI R “Membuat Histogram 3 Pilihan Warna”

Tampilan Awal

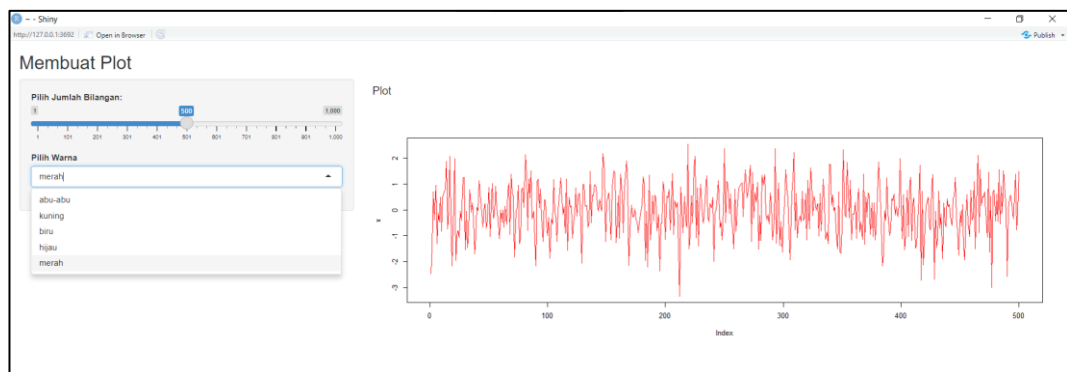


Tampilan Setelah Diatur (sesuai kebutuhan)



UJI COBA

Bagaimana jika membuat plot data bangkitan random normal dengan pilihan 5 warna dan nilai maksimum slider-nya 1000?



Keterangan penggunaan syntax plot:

```
## Default S3 method:  
plot(x, y = NULL, type = "p", xlim = NULL, ylim = NULL,  
     log = "", main = NULL, sub = NULL, xlab = NULL, ylab = NULL,  
     ann = par("ann"), axes = TRUE, frame.plot = axes,  
     panel.first = NULL, panel.last = NULL, asp = NA,  
     xgap.axis = NA, ygap.axis = NA,  
     ...)
```

Arguments

<code>x, y</code>	the <code>x</code> and <code>y</code> arguments provide the <code>x</code> and <code>y</code> coordinates for the plot. Any reasonable way of defining the coordinates is acceptable. See the function xy.coords for details. If supplied separately, they must be of the same length.
<code>type</code>	1-character string giving the type of plot desired. The following values are possible, for details, see plot : "p" for points, "l" for lines, "b" for both points and lines, "c" for empty points joined by lines, "o" for overplotted points and lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
<code>xlim</code>	the <code>x</code> limits (<code>x1</code> , <code>x2</code>) of the plot. Note that <code>x1 > x2</code> is allowed and leads to a 'reversed axis'. The default value, <code>NULL</code> , indicates that the range of the finite values to be plotted should be used.
<code>ylim</code>	the <code>y</code> limits of the plot.
<code>log</code>	a character string which contains "x" if the <code>x</code> axis is to be logarithmic, "y" if the <code>y</code> axis is to be logarithmic and "xy" or "yx" if both axes are to be logarithmic.
<code>main</code>	a main title for the plot, see also title .
<code>sub</code>	a sub title for the plot.
<code>xlab</code>	a label for the <code>x</code> axis, defaults to a description of <code>x</code> .
<code>ylab</code>	a label for the <code>y</code> axis, defaults to a description of <code>y</code> .
<code>ann</code>	a logical value indicating whether the default annotation (title and <code>x</code> and <code>y</code> axis labels) should appear on the plot.
<code>axes</code>	a logical value indicating whether both axes should be drawn on the plot. Use graphical parameter "xaxt" or "yaxt" to suppress just one of the axes.
<code>frame.plot</code>	a logical indicating whether a box should be drawn around the plot.
<code>panel.first</code>	an 'expression' to be evaluated after the plot axes are set up but before any plotting takes place. This can be useful for drawing background grids or scatterplot smooths. Note that this works by lazy evaluation: passing this argument from other <code>plot</code> methods may well not work since it may be evaluated too early.
<code>panel.last</code>	an expression to be evaluated after plotting has taken place but before the axes, title and box are added. See the comments about <code>panel.first</code> .
<code>asp</code>	the <code>y/x</code> aspect ratio, see plot.window .
<code>xgap.axis,</code> <code>ygap.axis</code>	the <code>x/y</code> axis gap factors, passed as <code>gap.axis</code> to the two axis () calls (when <code>axes</code> is true, as per default).
<code>...</code>	other graphical parameters (see par and section 'Details' below).