

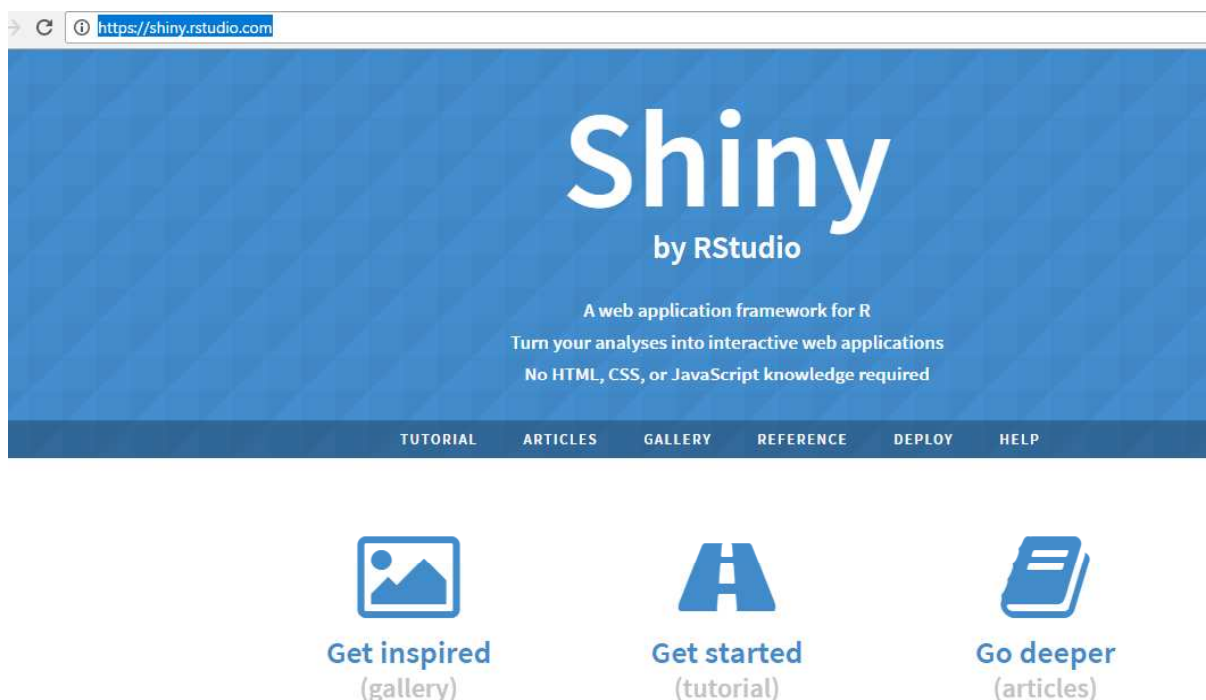
BAB 13

SEKILAS MEMBUAT APLIKASI OLAH DATA SEDERHANA DENGAN SHINY

13.1 Menginstal Package Shiny

Pada Bab 13 ini, akan diberikan contoh bagaimana membuat aplikasi olah data dengan shiny. Shiny adalah salah satu *package* R (*open source R package*) dari RStudio yang dapat digunakan untuk membuat *interactive web application*. Berikut alamat *website* resmi shiny <https://shiny.rstudio.com/>. Gambar 13.1 dan Gambar 13.2 disajikan cuplikan gambar dari *website* <https://shiny.rstudio.com/>.

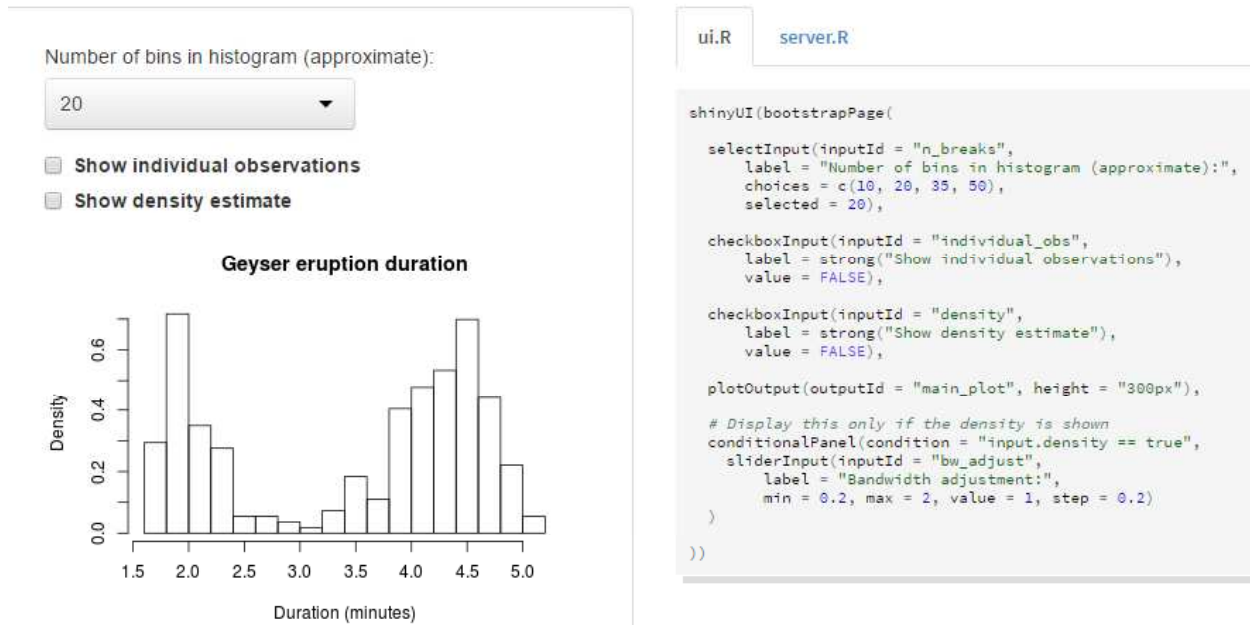
Untuk membuat program aplikasi dengan shiny, terlebih dahulu menginstal *package* shiny. Perhatikan Gambar 13.3. Pada Gambar 13.3 akan diinstal *package* shiny (**pastikan telah terkoneksi internet!!!**). Pada Gambar 13.4 diperlihatkan sedang proses menginstal *package* shiny. Pada Gambar 13.5 diperlihatkan telah sukses menginstal *package* shiny.



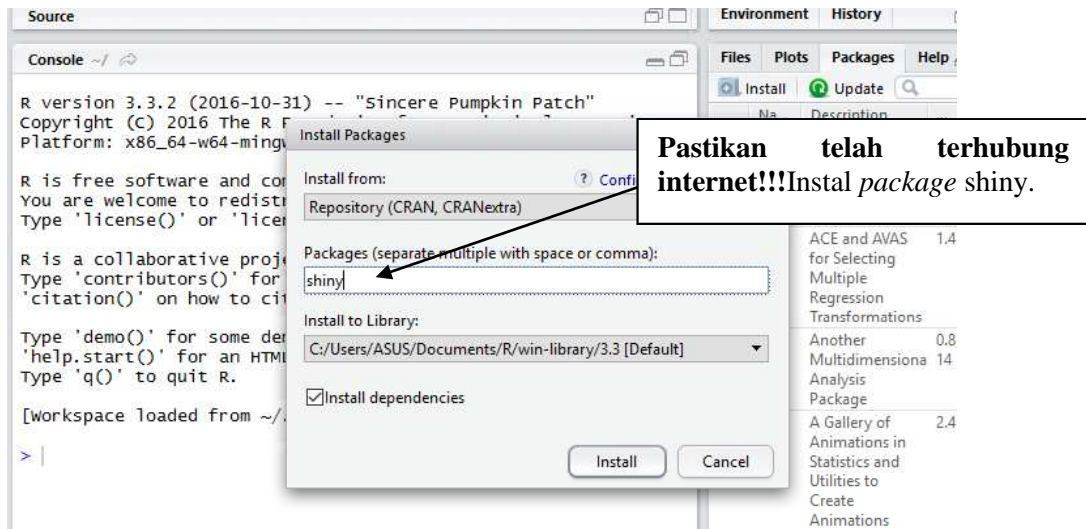
Gambar 13.1

Here is a Shiny app

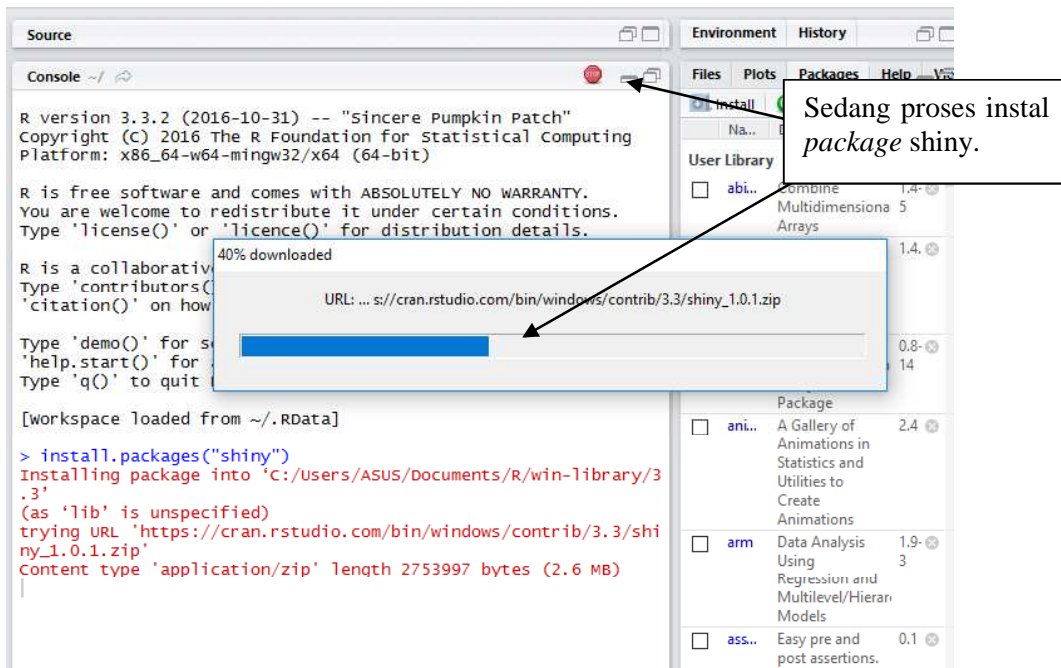
Shiny apps are easy to write. No web development skills are required.



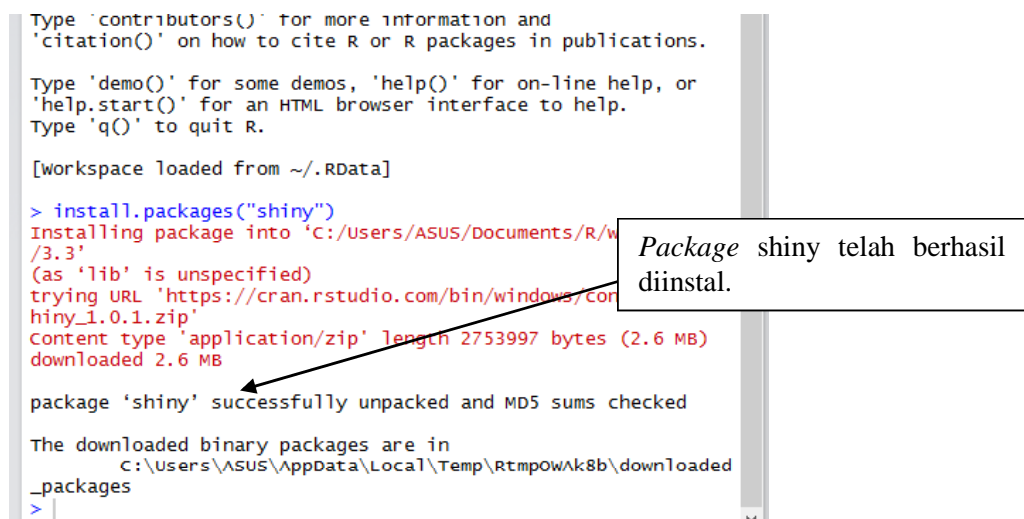
Gambar 13.2



Gambar 13.3



Gambar 13.4



Gambar 13.5

13.2 Membuat Program Aplikasi Shiny Paling Sederhana (Praktek 1)

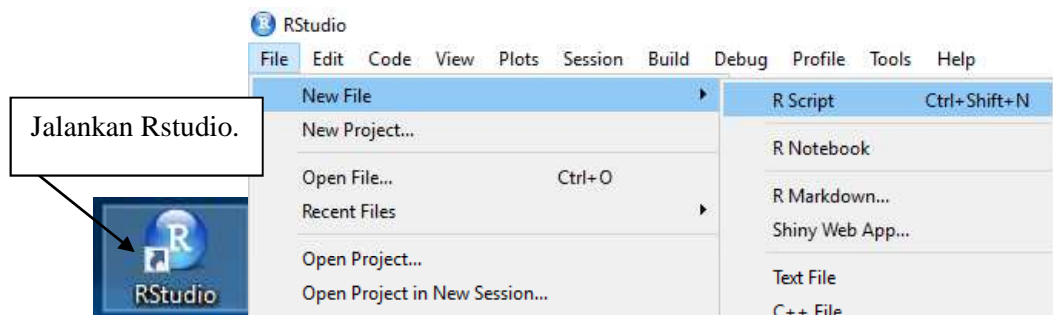
Pada Gambar 13.6 diperlihatkan program aplikasi shiny paling sederhana, yakni **tampilan kosong**. Sekarang, akan akan dijelaskan langkah demi langkah untuk membuat program aplikasi shiny tersebut.



Gambar 13.6 (Program Aplikasi Shiny Paling Sederhana, Tampilan Kosong)

Langkah Ke-1

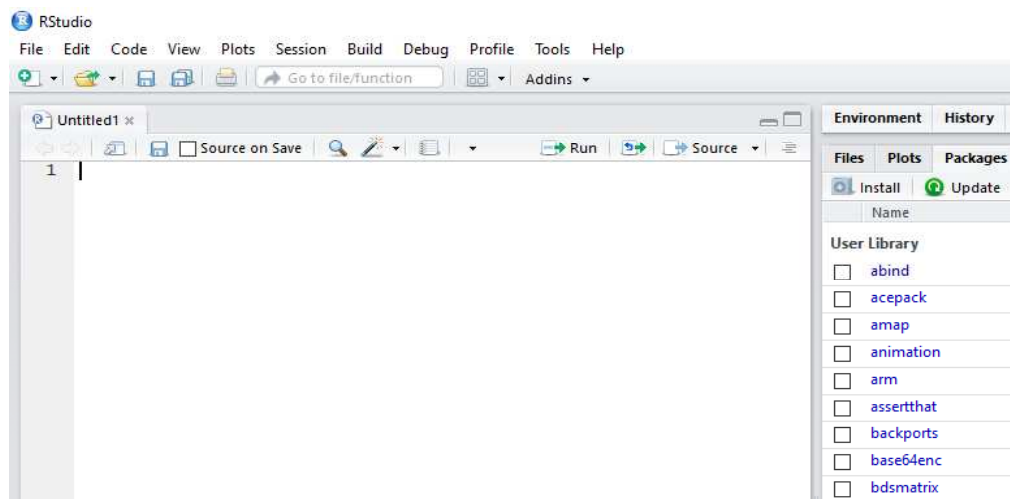
Jalankan Rstudio terlebih dahulu. Selanjutnya pilih *File =>New File =>R Script* (Gambar 13.7), sehingga muncul tampilan seperti pada Gambar 13.8.



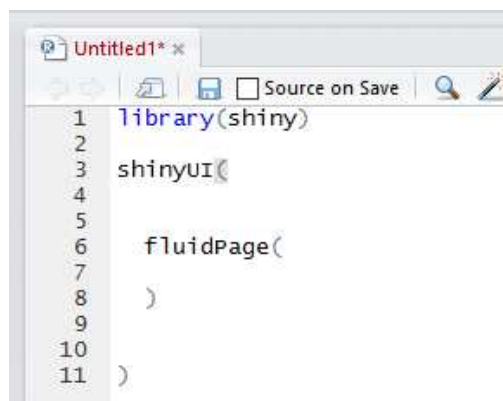
Gambar 13.7

Langkah Ke-2

Selanjutnya tuliskan kode R seperti pada Gambar 13.9.



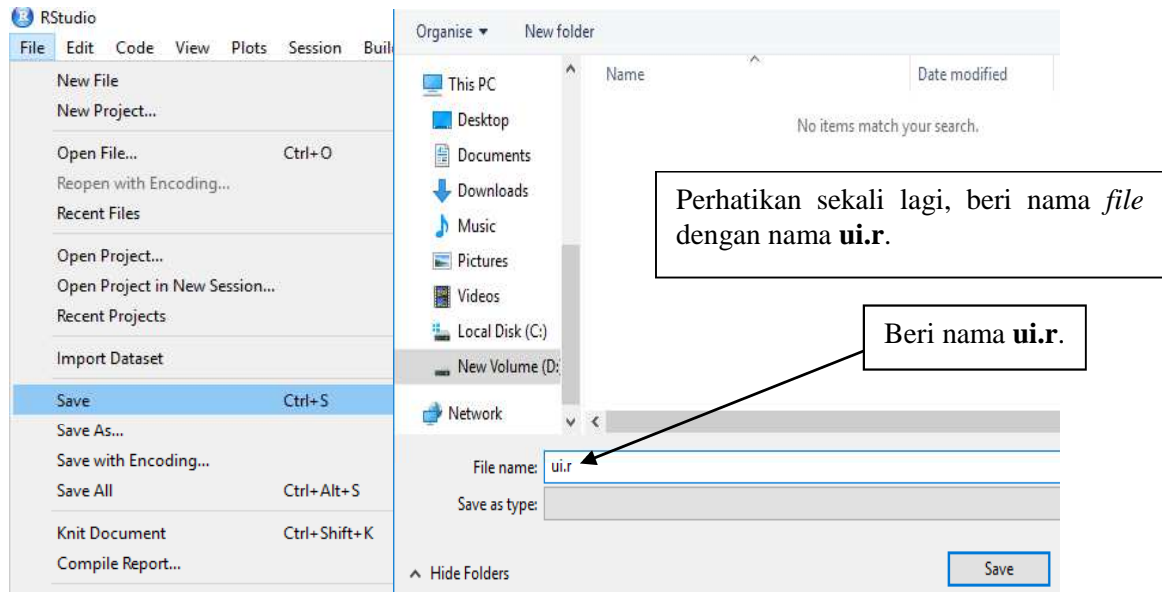
Gambar 13.8



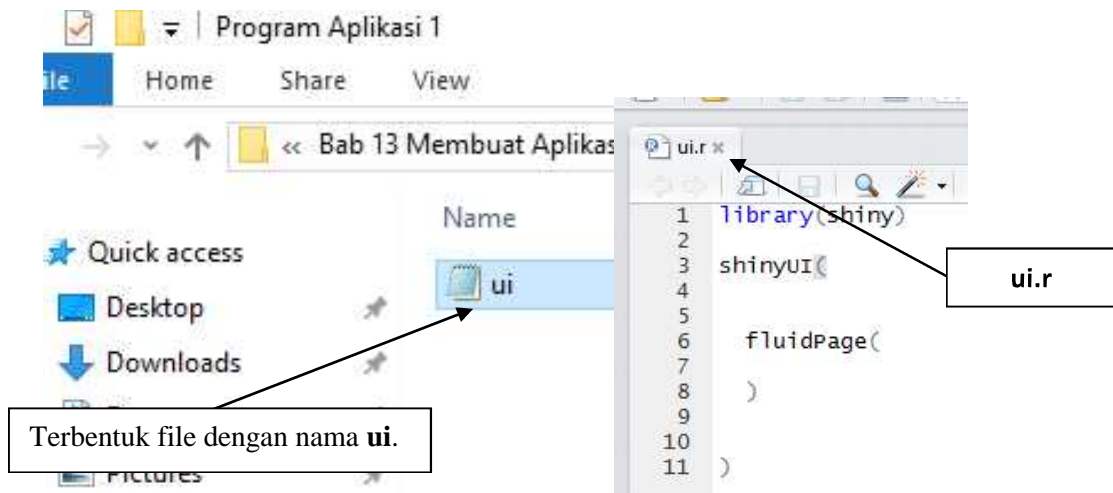
Gambar 13.9

Langkah Ke-3

Setelah kode R ditulis seperti pada Gambar 13.9, selanjutnya pilih *File* => *Save* (Gambar 13.10). Pada bagian *File name*, beri nama **ui.r**. Kemudian simpan.



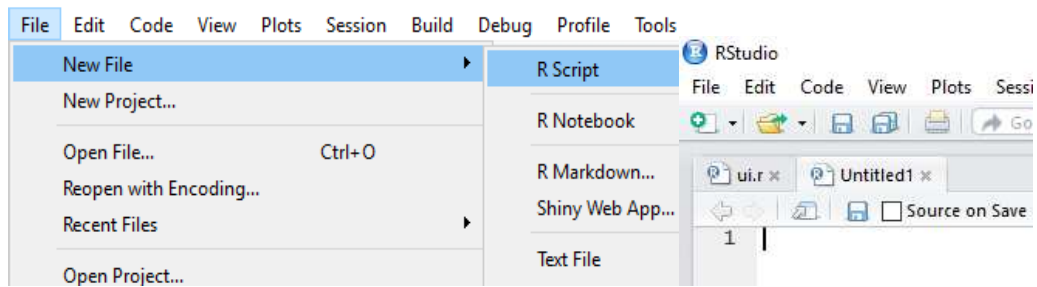
Gambar 13.10



Gambar 13.11

Langkah Ke-4

Pada langkah ke-3, telah dibuat *file* dengan nama **ui.r**. Sekarang pilih *File* => *New File* => *R Script* (Gambar 13.12), sehingga muncul tampilan seperti pada Gambar 13.12.



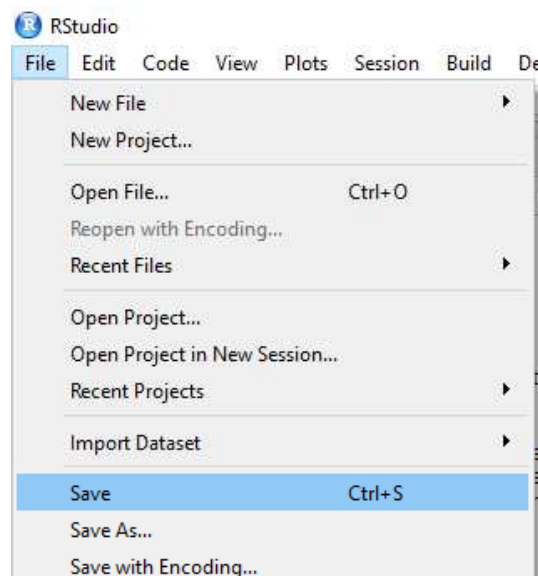
Gambar 13.12

Selanjutnya tuliskan kode R seperti pada Gambar 13.13.

```
1 library(shiny)
2
3 shinyserver (
4
5
6   function(input, output) {
7
8   }
9 }
10 )
```

Gambar 13.13

Setelah kode R ditulis seperti pada Gambar 13.13, selanjutnya pilih *File* => *Save* (Gambar 13.14). Pada bagian *File name*, beri nama **server.r**. Kemudian simpan.

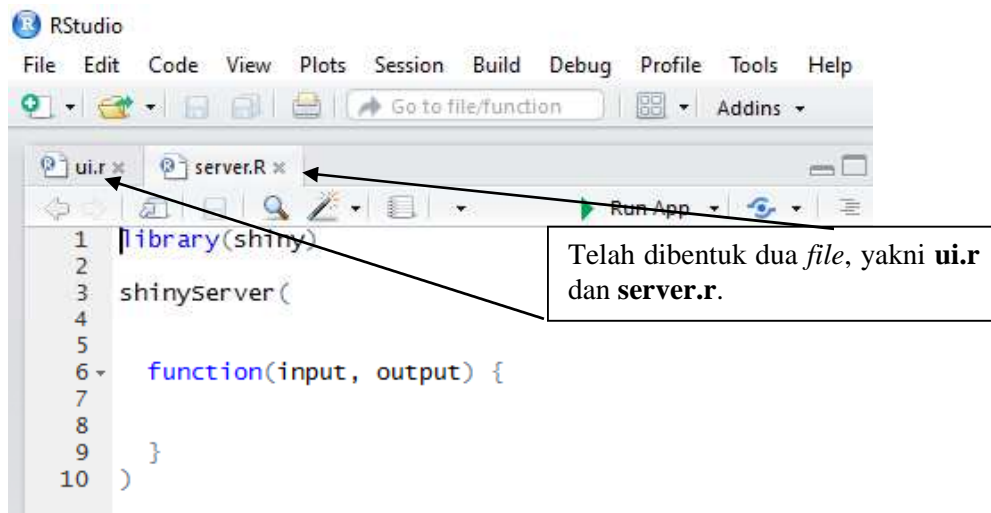


Gambar 13.14



Gambar 13.15

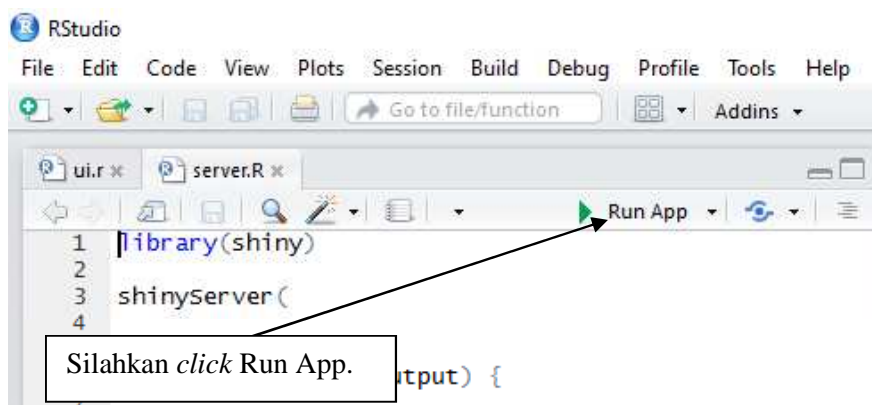
Perhatikan bahwa telah dibentuk dua buah *file* dengan nama **ui** dan **server** (Gambar 13.15).



Gambar 13.16

Langkah Ke-5 (Jalankan Aplikasi)

Langkah terakhir, jalankan program aplikasi dengan meng-*click* **Run App** (Gambar 13.17). Maka hasilnya seperti pada Gambar 13.18.



Gambar 13.17



Gambar 13.18 (Program Aplikasi Shiny Paling Sederhana, Tampilan Kosong)

Penjelasan Kode `ui.r` dan `server.r`

Ada dua komponen untuk membentuk program aplikasi dengan shiny, yakni:

- ⇒ *user-interface script.*
- ⇒ *server script.*

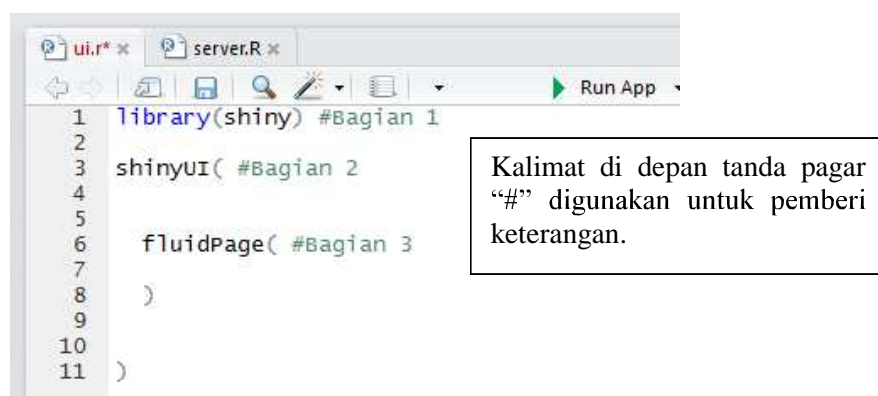
Pada proses pembuatan program aplikasi shiny, telah dibuat *file* dengan nama `ui.r` dan `server.r`. Perhatikan bahwa “ui” singkatan dari *user-interface*. Pada kode R pada bagian *user-interface*(`ui.r`), bertujuan untuk **mendesain tampilan dari program aplikasi**. *File ui.r* dan *server.r* disimpan dalam **folder yang sama**. Berikut kutipan dari halaman *website* <https://shiny.rstudio.com/tutorial/lesson1/>.

“*Shiny apps have two components:*

a user-interface script
a server script

The user-interface (ui) script controls the layout and appearance of your app. It is defined in a source script named ui.R”

Gambar 13.19 disajikan kode R untuk `ui.r` yang paling sederhana (tampilan kosong).



Gambar 13.19

Berdasarkan Gambar 13.19:

- ⇒ Kode R bagian 1 bertujuan untuk mengaktifkan (*load*) *package* shiny. Hal ini perlu dilakukan karena akan dibuat program aplikasi dengan menggunakan shiny.

- ⇒ Kode R bagian 2 merupakan format dari shiny untuk *user-interface* di shiny.
- ⇒ Pada contoh selanjutnya, di dalam fungsi *fluidPage*, tampilan dari program aplikasi shiny diatur.

Berikut kutipan dari halaman *website* <https://shiny.rstudio.com/tutorial/lesson2/>.

“Shiny *ui.R* scripts use the function *fluidPage* to create a display that automatically adjusts to the dimensions of your user’s browser window. **You lay out your app by placing elements in the *fluidPage* function.**”

For example, the *ui.R* script below creates a user-interface that has a title panel and then a sidebar layout, which includes a sidebar panel and a main panel. Note that these elements are placed within the *fluidPage* function.”

```

RStudio
File Edit Code View Plots Session Build Debug Profile
Go to file/function
ui.r* x server.R x
Run App
1 library(shiny) #Bagian 1
2
3 shinyserver( #Bagian 2
4
5
6 function(input, output) { #Bagian 3
7
8
9 }
10 )

```

Gambar 13.20

Pada Gambar 13.20 disajikan kode R untuk **server.r**. Berdasarkan Gambar 13.20, diketahui:

- ⇒ Kode R bagian 1 bertujuan untuk mengaktifkan (*load*) *package* shiny. Hal ini perlu dilakukan karena akan dibuat program aplikasi dengan menggunakan shiny.
- ⇒ Kode R bagian 2 merupakan format dari shiny untuk server di shiny.
- ⇒ Pada contoh selanjutnya, di dalam fungsi *function(input,output)* nantinya berisi instruksi-instruksi mengenai, misalkan instruksi-instruksi untuk perhitungan, penyajian data ke dalam grafik, dan sebagainya.

Berikut kutipan dari halaman *website* <https://shiny.rstudio.com/tutorial/lesson1/>.

“The *server.R* script contains the **instructions that your computer needs to build your app**. Here is the *server.R*”

13.2 Membuat Program Aplikasi Penjumlahan (Praktek 2)

Pada Gambar 13.21 dan Gambar 13.22 diperlihatkan program aplikasi yang dapat digunakan untuk melakukan penjumlahan dua buah bilangan. Pada program aplikasi tersebut, diminta untuk memberikan dua buah bilangan, yakni bilangan pertama dan bilangan kedua. Setelah diinput bilangan pertama dan bilangan kedua, maka akan secara otomatis muncul hasil penjumlahan dari dua buah bilangan tersebut.

Program Aplikasi untuk Penjumlahan Dua Bilangan

Silahkan Berikan Bilangan Pertama:

Silahkan Berikan Bilangan Kedua:

Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2

Gambar 13.21

Program Aplikasi untuk Penjumlahan Dua Bilangan

Silahkan Berikan Bilangan Pertama:

Silahkan Berikan Bilangan Kedua:

Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2

Gambar 13.22

Untuk membuat program aplikasi shiny tersebut, tulis kode R untuk **ui.r** dan **server.r** seperti pada Gambar 13.23 dan 13.24. Simpan **ui.r** dan **server.r** dalam satu folder. Kemudian jalankan aplikasi dengan meng-*click* **Run App**, seperti pada Gambar 13.25. Akan muncul program aplikasi shiny seperti pada Gambar 13.26.

```

1 library(shiny)
2
3 shinyUI(
4
5
6   fluidPage(
7
8     headerPanel("Program Aplikasi untuk Penjumlahan Dua Bilangan"),
9
10    numericInput("bilangan1", label="Silahkan Berikan Bilangan Pertama:", value=""),
11    numericInput("bilangan2", label="Silahkan Berikan Bilangan kedua:", value=""),
12
13    h3("Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2"),
14    verbatimTextOutput("hitung")
15  )
16 )
17
18
19
20 )

```

Perhatian!!! Harap teliti dalam pengetikan kode R terhadap tanda titik, koma, buka kurung kurawal.

Gambar 13.23 (Kode R untuk ui.r)

```

1 library(shiny)
2
3 shinyServer(
4
5
6   function(input, output) {
7
8
9     output$hitung <- renderPrint({ #proses perhitungan pada hitung
10
11     simpan_bilangan1 <- input$bilangan1 #bagian 1
12     simpan_bilangan2 <- input$bilangan2 #bagian 1
13     jumlah = simpan_bilangan1 + simpan_bilangan2 #bagian 2
14
15     cat(sprintf("%F", jumlah)) #bagian 3
16
17   })
18
19 }
20
21 )

```

Perhatian!!! Harap teliti dalam pengetikan kode R terhadap tanda titik, koma, buka kurung kurawal.

Gambar 13.24 (Kode R untuk server.r)



Gambar 13.25

Program Aplikasi untuk Penjumlahan Dua Bilangan

Silahkan Berikan Bilangan Pertama:

Silahkan Berikan Bilangan Kedua:

Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2

Input bilangan sesuai yang diinginkan.

Gambar 13.26

Program Aplikasi untuk Penjumlahan Dua Bilangan

Silahkan Berikan Bilangan Pertama:

Silahkan Berikan Bilangan Kedua:

Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2

Gambar 13.27

Penjelasan Kode ui.R

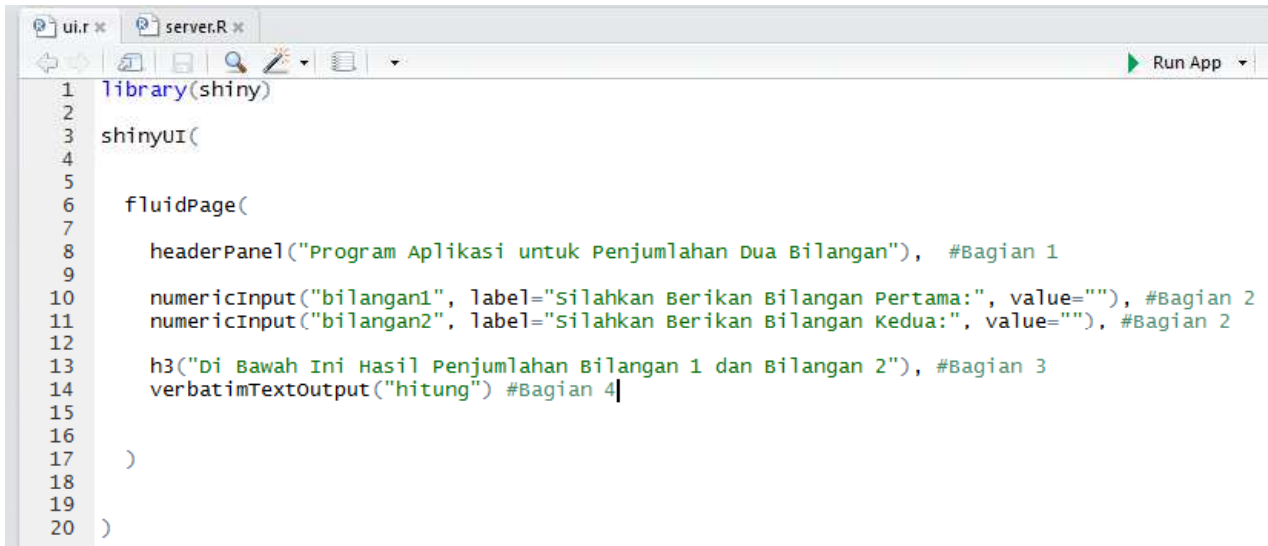
Ada dua komponen untuk membentuk program aplikasi dengan shiny, yakni:

- ⇒ *user-interface script.*
- ⇒ *server script.*

Pada proses pembuatan program aplikasi untuk penjumlahan dua bilangan, telah dibuat *file* dengan nama **ui.r** dan **server.r**. Perhatikan bahwa “ui” singkatan dari *user-interface*. Pada kode R pada bagian *user-interface*, bertujuan untuk **mendesain tampilan dari program aplikasi**.

Berdasarkan Gambar 13.28:

- ⇒ Pada bagian 1 bertujuan untuk membuat judul “Program Aplikasi untuk Penjumlahan Dua Bilangan”.
- ⇒ Pada bagian 2, diminta untuk menginput bilangan, yakni bilangan pertama dan bilangan kedua.
- ⇒ Pada bagian 3 akan mencetak tulisan “Di Bawah ini Hasil Penjumlahan 1 dan Bilangan 2”.
- ⇒ Pada bagian 4 akan menampilkan hasil dari **hitung**, yang mana proses perhitungan dari **hitung** dapat dilihat pada *file server.r*.



```
1 library(shiny)
2
3 shinyUI(
4
5
6   fluidPage(
7
8     headerPanel("Program Aplikasi untuk Penjumlahan Dua Bilangan"), #Bagian 1
9
10    numericInput("bilangan1", label="Silahkan Berikan Bilangan Pertama:", value=""), #Bagian 2
11    numericInput("bilangan2", label="Silahkan Berikan Bilangan Kedua:", value=""), #Bagian 2
12
13    h3("Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2"), #Bagian 3
14    verbatimTextOutput("hitung") #Bagian 4|
15
16  )
17 )
18
19
20 )
```

Gambar 13.28

Penjelasan Kode server.R

Berdasarkan Gambar 13.29:

⇒ Pada bagian 1, kode R

Simpan_bilangan1 <- input\$bilangan1

dapat diartikan **simpan_bilangan1** ditugaskan untuk **menyimpan bilangan** yang diinput pada **bilangan1**. Kode R

Simpan_bilangan2 <- input\$bilangan2

dapat diartikan **simpan_bilangan2** ditugaskan untuk **menyimpan bilangan** yang diinput pada **bilangan2**.

⇒ Pada bagian 2, kode R

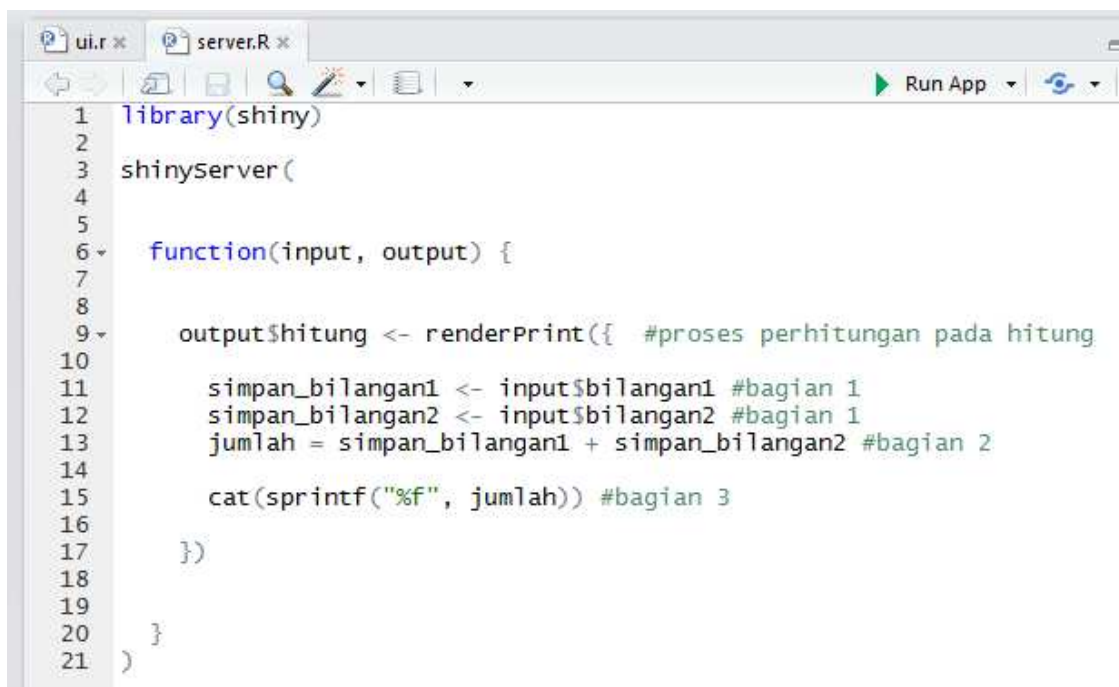
jumlah = simpan_bilangan1 + simpan_bilangan2

dapat diartikan **jumlah** ditugaskan **untuk menyimpan bilangan dari hasil penjumlahan** antara **simpan_bilangan1** dan **simpan_bilangan2**.

⇒ Pada bagian 3, kode R

cat(sprintf("%f", jumlah))

dapat diartikan untuk menampilkan bilangan pecahan (%f menandakan untuk bilangan pecahan) pada jumlah.

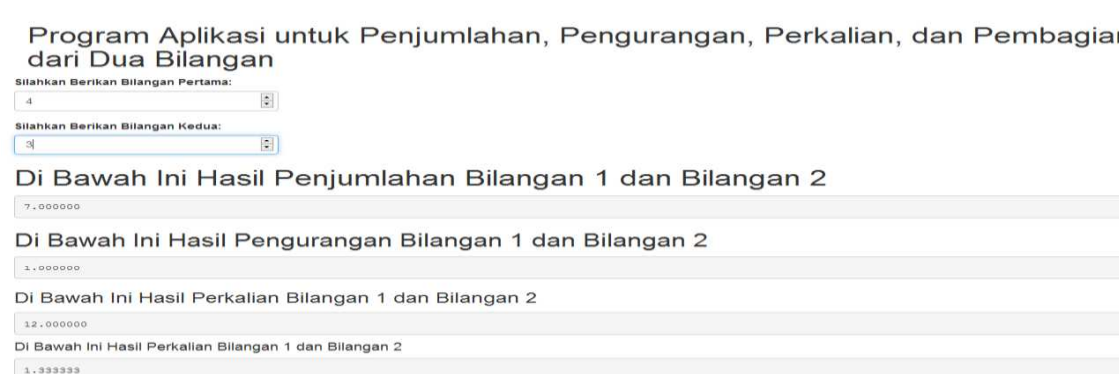


```
1 library(shiny)
2
3 shinyServer(
4
5
6 function(input, output) {
7
8
9   output$hitung <- renderPrint({ #proses perhitungan pada hitung
10
11     simpan_bilangan1 <- input$bilangan1 #bagian 1
12     simpan_bilangan2 <- input$bilangan2 #bagian 1
13     jumlah = simpan_bilangan1 + simpan_bilangan2 #bagian 2
14
15     cat(sprintf("%f", jumlah)) #bagian 3
16
17   })
18
19 }
20
21 )
```

Gambar 13.29

13.3 Membuat Program Aplikasi Penjumlahan, Pengurangan, Perkalian, dan Pembagian (Praktek 3)

Program aplikasi shiny praktek 3 ini, merupakan pengembangan dari program aplikasi shiny pada praktek 2, yang mana selain penjumlahan, juga dapat melakukan pengurangan, perkalian, dan pembagian. Perhatikan Gambar 13.30 dan Gambar 13.31. Pada program aplikasi tersebut, diminta untuk memberikan dua buah bilangan, yakni bilangan pertama dan bilangan kedua. Setelah diinput bilangan pertama dan bilangan kedua, maka akan secara otomatis muncul hasil penjumlahan, pengurangan, perkalian, dan pembagian dari dua buah bilangan tersebut.



Gambar 13.30

Program Aplikasi untuk Penjumlahan, Pengurangan, Perkalian, dan Pembagian dari Dua Bilangan

Silahkan Berikan Bilangan Pertama:

10

Silahkan Berikan Bilangan Kedua:

2

Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2

12.000000

Di Bawah Ini Hasil Pengurangan Bilangan 1 dan Bilangan 2

8.000000

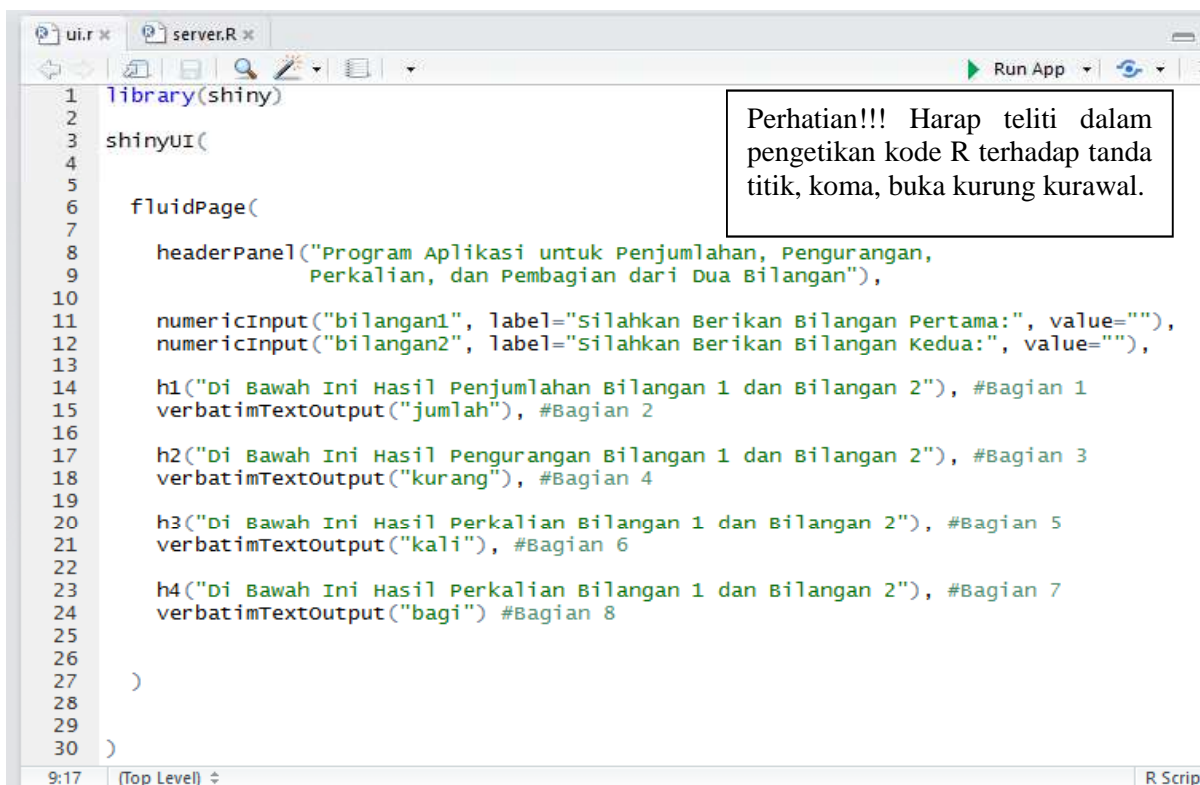
Di Bawah Ini Hasil Perkalian Bilangan 1 dan Bilangan 2

20.000000

Di Bawah Ini Hasil Pembagian Bilangan 1 dan Bilangan 2

5.000000

Gambar 13.31



```
1 library(shiny)
2
3 shinyUI(
4
5
6   fluidPage(
7
8     headerPanel("Program Aplikasi untuk Penjumlahan, Pengurangan,
9                 Perkalian, dan Pembagian dari Dua Bilangan"),
10
11    numericInput("bilangan1", label="Silahkan Berikan Bilangan Pertama:", value=""),
12    numericInput("bilangan2", label="Silahkan Berikan Bilangan Kedua:", value=""),
13
14    h1("Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2"), #Bagian 1
15    verbatimTextOutput("jumlah"), #Bagian 2
16
17    h2("Di Bawah Ini Hasil Pengurangan Bilangan 1 dan Bilangan 2"), #Bagian 3
18    verbatimTextOutput("kurang"), #Bagian 4
19
20    h3("Di Bawah Ini Hasil Perkalian Bilangan 1 dan Bilangan 2"), #Bagian 5
21    verbatimTextOutput("kali"), #Bagian 6
22
23    h4("Di Bawah Ini Hasil Pembagian Bilangan 1 dan Bilangan 2"), #Bagian 7
24    verbatimTextOutput("bagi") #Bagian 8
25
26  )
27
28
29
30 )
```

Perhatian!!! Harap teliti dalam pengetikan kode R terhadap tanda titik, koma, buka kurung kurawal.

Gambar 13.32 (Kode R untuk ui.r)


```

1 library(shiny)
2
3 shinyserver(
4
5
6 function(input, output) {
7
8 #####
9 #####Penjumlahan#####
10 #####
11
12 output$jumlah <- renderPrint({
13
14     simpan_bilangan1 <- input$bilangan1
15     simpan_bilangan2 <- input$bilangan2
16     jumlah = simpan_bilangan1 + simpan_bilangan2
17     cat(sprintf("%f", jumlah))
18
19 })
20
21 #####
22 #####Pengurangan#####
23 #####
24
25 output$kurang <- renderPrint({
26
27     simpan_bilangan1 <- input$bilangan1
28     simpan_bilangan2 <- input$bilangan2
29     selisih = simpan_bilangan1 - simpan_bilangan2
30     cat(sprintf("%f", selisih))
31
32 })
33
34

```

Perhatian!!! Harap teliti dalam pengetikan kode R terhadap tanda titik, koma, buka kurung kurawal.

Gambar 13.33 (Kode R untuk server.r)

```

29     selisih = simpan_bilangan1 - simpan_bilangan2
30     cat(sprintf("%f", selisih))
31
32 })
33
34
35 #####
36 #####Perkalian#####
37 #####
38
39 output$kali <- renderPrint({
40
41     simpan_bilangan1 <- input$bilangan1
42     simpan_bilangan2 <- input$bilangan2
43     perkalian = simpan_bilangan1 * simpan_bilangan2
44     cat(sprintf("%f", perkalian))
45
46 })
47
48
49 #####
50 #####Pembagian#####
51 #####
52
53 output$bagi <- renderPrint({
54
55     simpan_bilangan1 <- input$bilangan1
56     simpan_bilangan2 <- input$bilangan2
57     dibagi = simpan_bilangan1 / simpan_bilangan2
58     cat(sprintf("%f", dibagi))
59
60 })
61
62 }
63 )

```

Gambar 13.34 (Sambungan Kode R untuk server.r)

Untuk membuat program aplikasi shiny tersebut, tulis kode R untuk **ui.r** dan **server.r** seperti pada Gambar 13.32 dan 13.34. Simpan **ui.r** dan **server.r** dalam satu folder. Kemudian jalankan aplikasi dengan meng-*click* **Run App**. Akan muncul program aplikasi shiny seperti pada Gambar 13.30 dan Gambar 13.31.

Berdasarkan Gambar 13.35, perhatikan bahwa `h1("kalimat")`, `h2("kalimat")`, `h3("kalimat")`, dan `h4("kalimat")` bertujuan untuk menampilkan kalimat di antara tanda petik ganda, dengan ukuran kalimat yang berbeda-beda.

- ⇒ Pada bagian 2 akan menampilkan hasil dari **jumlah**, yang mana proses perhitungan dari **jumlah** dapat dilihat pada *file server.r*.
- ⇒ Pada bagian 4 akan menampilkan hasil dari **kurang**, yang mana proses perhitungan dari **kurang** dapat dilihat pada *file server.r*.
- ⇒ Pada bagian 6 akan menampilkan hasil dari **kali**, yang mana proses perhitungan dari **kali** dapat dilihat pada *file server.r*.
- ⇒ Pada bagian 8 akan menampilkan hasil dari **bagi**, yang mana proses perhitungan dari **bagi** dapat dilihat pada *file server.r*.

```

1 library(shiny)
2
3 shinyUI(
4
5
6   fluidPage(
7
8     headerPanel("Program Aplikasi untuk Penjumlahan, Pengurangan,
9                 Perkalian, dan Pembagian dari Dua Bilangan"),
10
11    numericInput("bilangan1", label="Silahkan Berikan Bilangan Pertama:", value=""),
12    numericInput("bilangan2", label="Silahkan Berikan Bilangan Kedua:", value=""),
13
14    h1("Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2"), #Bagian 1
15    verbatimTextoutput("jumlah"), #Bagian 2
16
17    h2("Di Bawah Ini Hasil Pengurangan Bilangan 1 dan Bilangan 2"), #Bagian 3
18    verbatimTextoutput("kurang"), #Bagian 4
19
20    h3("Di Bawah Ini Hasil Perkalian Bilangan 1 dan Bilangan 2"), #Bagian 5
21    verbatimTextoutput("kali"), #Bagian 6
22
23    h4("Di Bawah Ini Hasil Pembagian Bilangan 1 dan Bilangan 2"), #Bagian 7
24    verbatimTextoutput("bagi") #Bagian 8
25
26  )
27
28
29
30 )

```

Gambar 13.35

Berikut disajikan bagian dari kode R pada **server.r**.

```

output$kali <- renderPrint({

  simpan_bilangan1 <- input$bilangan1

```

```

simpan_bilangan2 <- input$bilangan2
perkalian = simpan_bilangan1 * simpan_bilangan2
cat(sprintf("%f", perkalian))

})

```

Berdasarkan kode R di atas,

```

Simpan_bilangan1 <- input$bilangan1

```

dapat diartikan **simpan_bilangan1** ditugaskan untuk **menyimpan bilangan** yang diinput pada **bilangan1**. Kode R

```

Simpan_bilangan2 <- input$bilangan2

```

dapat diartikan **simpan_bilangan2** ditugaskan untuk **menyimpan bilangan** yang diinput pada **bilangan2**. Kode R

```

perkalian = simpan_bilangan1 * simpan_bilangan2

```

dapat diartikan **perkalian** ditugaskan untuk **menyimpan bilangan dari hasil perkalian** antara **simpan_bilangan1** dan **simpan_bilangan2**. Kode R

```

cat(sprintf("%f", perkalian))

```

dapat diartikan untuk menampilkan bilangan pecahan (%f menandakan untuk bilangan pecahan) pada **perkalian**.

13.4 Membuat Program Aplikasi Penjumlahan dengan Tampilan Berbeda (Praktek 4)

Tampilan program aplikasi shiny pada praktek 2 sebagai berikut (Gambar 13.36).

Program Aplikasi untuk Penjumlahan Dua Bilangan

Silahkan Berikan Bilangan Pertama:
2

Silahkan Berikan Bilangan Kedua:
3

Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2

5.000000

Gambar 13.36

Sekarang akan dibuat program aplikasi shiny seperti pada Gambar 13.37.

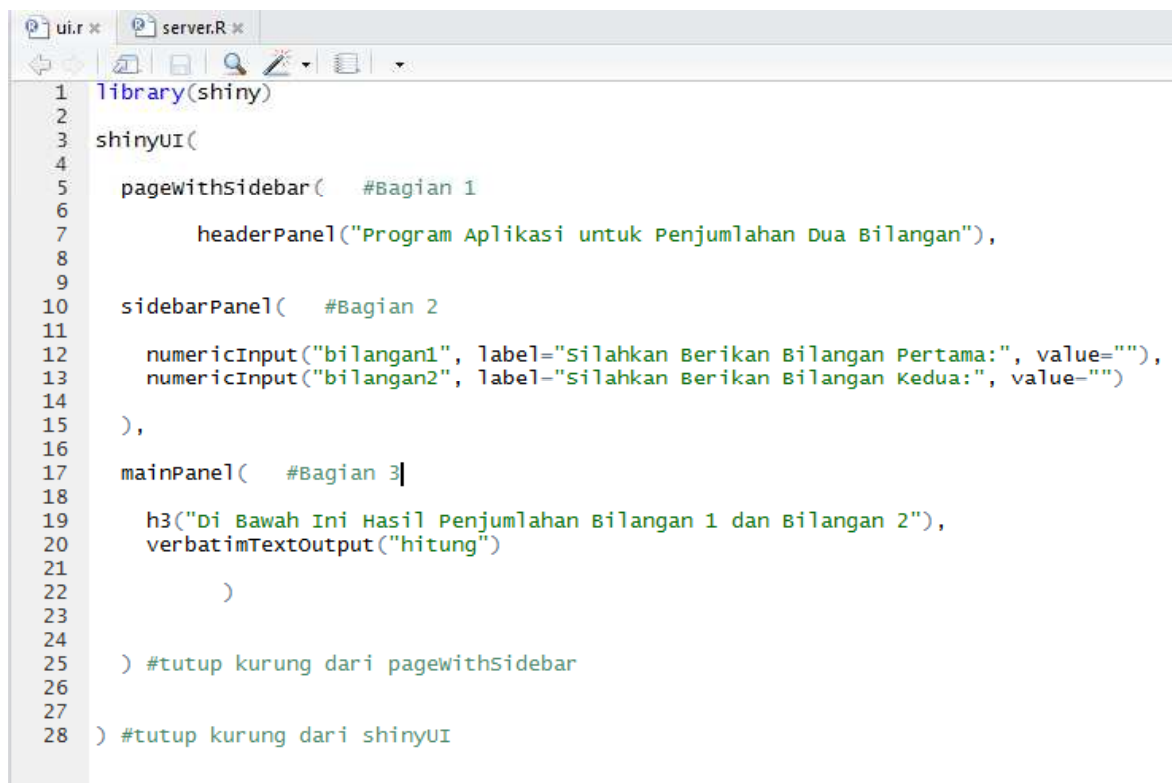
Program Aplikasi untuk Penjumlahan Dua Bilangan



The screenshot shows a web application interface. On the left, there are two input fields. The first is labeled "Silahkan Berikan Bilangan Pertama:" and contains the number "2". The second is labeled "Silahkan Berikan Bilangan Kedua:" and contains the number "4". On the right, there is a text area with the heading "Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2" and a result box containing "6.000000".

Gambar 13.37

Untuk membuat program aplikasi shiny tersebut, tulis kode R untuk **ui.r** dan **server.r** seperti pada Gambar 13.38 dan 13.39. Simpan **ui.r** dan **server.r** dalam satu folder. Kemudian jalankan aplikasi dengan meng-*click* **Run App**.



```
1 library(shiny)
2
3 shinyUI(
4   pagewithsidebar( #Bagian 1
5     headerPanel("Program Aplikasi untuk Penjumlahan Dua Bilangan"),
6
7     sidebarPanel( #Bagian 2
8       numericInput("bilangan1", label="Silahkan Berikan Bilangan Pertama:", value=""),
9       numericInput("bilangan2", label="Silahkan Berikan Bilangan Kedua:", value="")
10    ),
11
12    mainPanel( #Bagian 3|
13      h3("Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2"),
14      verbatimTextOutput("hitung")
15    )
16  )
17 ) #tutup kurung dari pagewithsidebar
18
19 ) #tutup kurung dari shinyUI
```

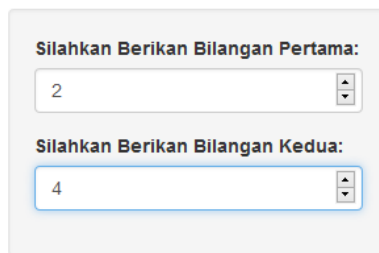
Gambar 13.38

Berdasarkan Gambar 13.38, pada bagian 1, yang sebelumnya **fluidPage** sekarang menjadi **sidebarPanel**. Pada bagian 2, **sidebarPanel** untuk tampilan pada bagian kiri (perhatikan Gambar 13.40). Sementara bagian 3, **mainPanel** untuk tampilan pada bagian kanan (perhatikan Gambar 13.41).

```
ui.r x server.R x
1 library(shiny)
2
3 shinyServer(
4
5
6 function(input, output) {
7
8
9 output$hitung <- renderPrint({ #proses perhitungan pada hitung
10
11   simpan_bilangan1 <- input$bilangan1 #bagian 1
12   simpan_bilangan2 <- input$bilangan2 #bagian 1
13   jumlah = simpan_bilangan1 + simpan_bilangan2 #bagian 2
14
15   cat(sprintf("%f", jumlah)) #bagian 3
16
17 })
18
19 }
20
21 )
```

Gambar 13.39

Program Aplikasi u



Gambar 13.40 (Bagian sidebarPanel)

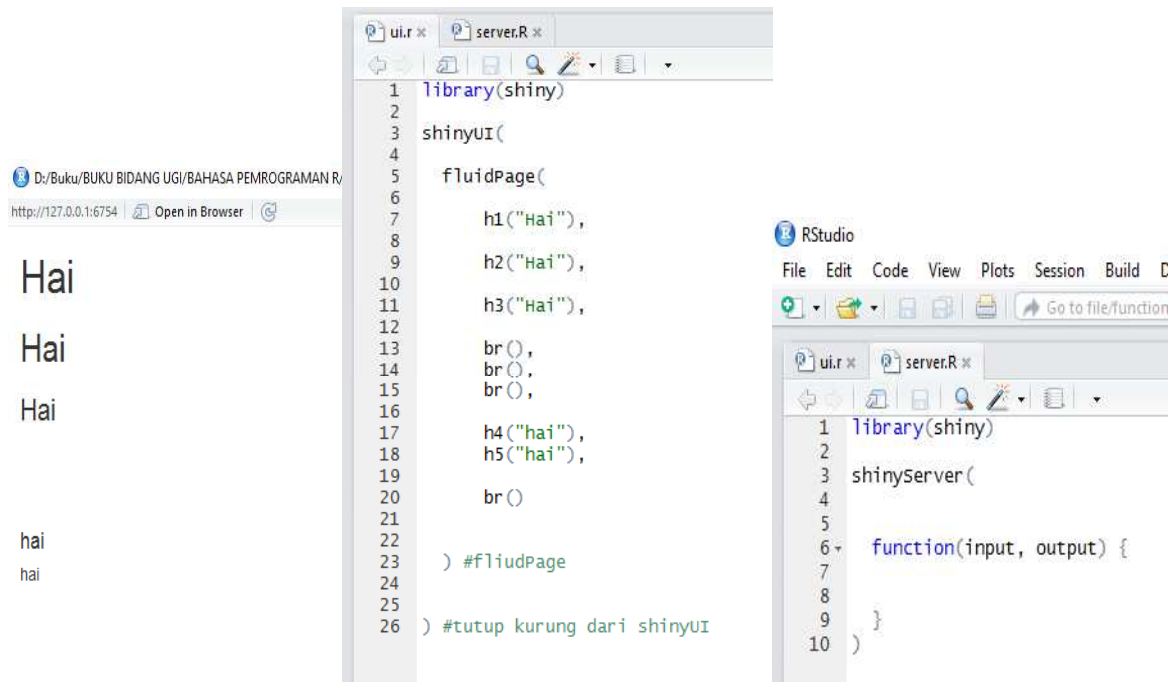
Di Bawah Ini Hasil Penjumlahan Bilangan 1 dan Bilangan 2

5.000000

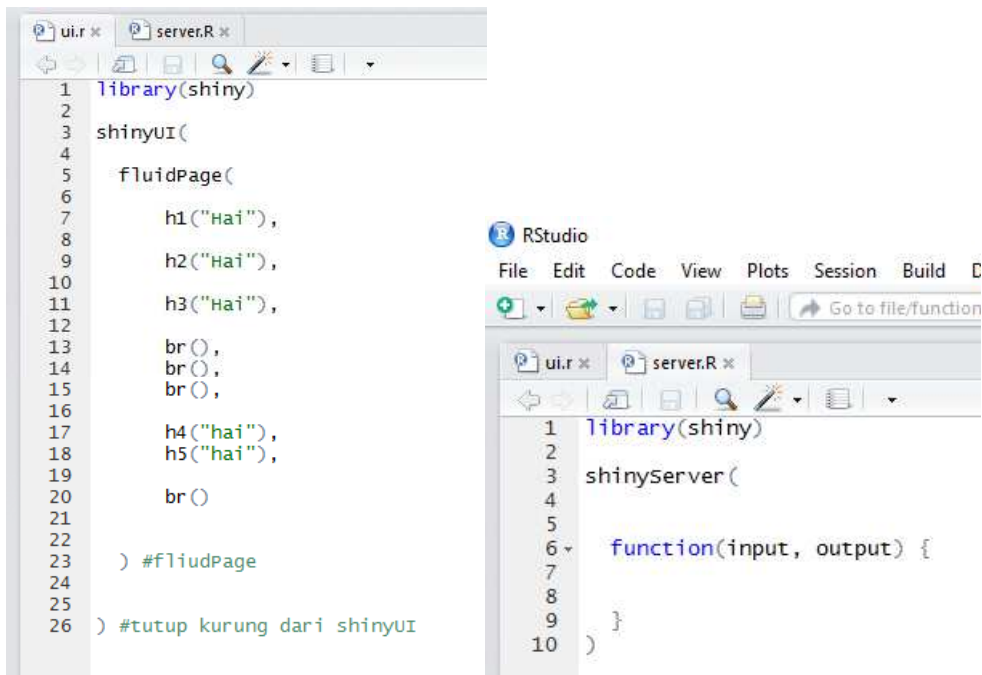
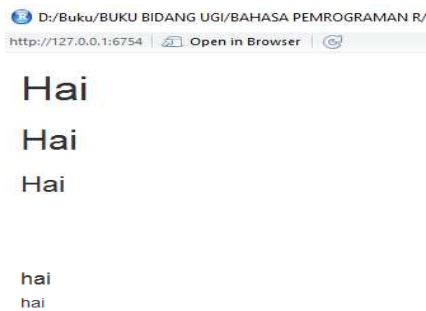
Gambar 13.41 (Bagian mainPanel)

13.5 Berbagai Jenis Style Tulisan (Praktek 5)

Berikut disajikan berbagai jenis *style* tulisan dalam shiny.



Gambar 13.42



Gambar 13.43

Hai

Hai

Hai

Prana Ugiana Gio

Prana Ugiana Gio

Hai Prana Ugiana Gio & Ronnie

Prana Ugiana Gio Prana Ugiana Gio Prana Ugiana Gio

Prana Ugiana Gio, lahir di Tanjung Pinang

```
1 library(shiny)
2
3 shinyUI(
4   fluidPage(
5     h1("Hai", style = "color:red;"),
6     h1("Hai", style = "color:red; text-align:center;"),
7     h1("Hai", style = "color:red; text-align:right;"),
8
9     h1("Prana Ugiana Gio",
10      style = "font-family: 'Lobster', cursive;
11      font-weight: 500; line-height: 1.1;
12      color: red;"),
13
14     h1("Prana Ugiana Gio",
15      style = "font-family: 'Lobster', cursive;
16      color: green; text-align:center;"),
17
18     h1( span("Hai", style = "color:red;"), "Prana Ugiana",
19        span("Gio", style="color:green;"), "&",
20        span("Ronnie", style="font-style:italic; color:blue;"),
21        style="text-align:center" ),
22
23     h1(span("Prana Ugiana Gio", style="font-size:10px"),
24        span("Prana Ugiana Gio", style="font-size:20px"),
25        span("Prana Ugiana Gio", style="font-size:30px" ) ),
26
27     p("Prana Ugiana Gio, ", span("lahir di Tanjung Pinang",
28        style="font-weight:bold; font-style:italic;") ),
29
30     br()
31   ) #akhir fluidPage
32 ) #tutup kurung dari shinyUI
```

```
1 library(shiny)
2
3 shinyServer(
4   function(input, output) {
5
6   }
7 )
```

Gambar 13.44

Prana Ugiana Gio

Prana Ugiana Gio

Prana Ugiana Gio

Prana Ugiana Gio

Prana Ugiana Gio

Prana Ugiana Gio

Prana Ugiana Gio

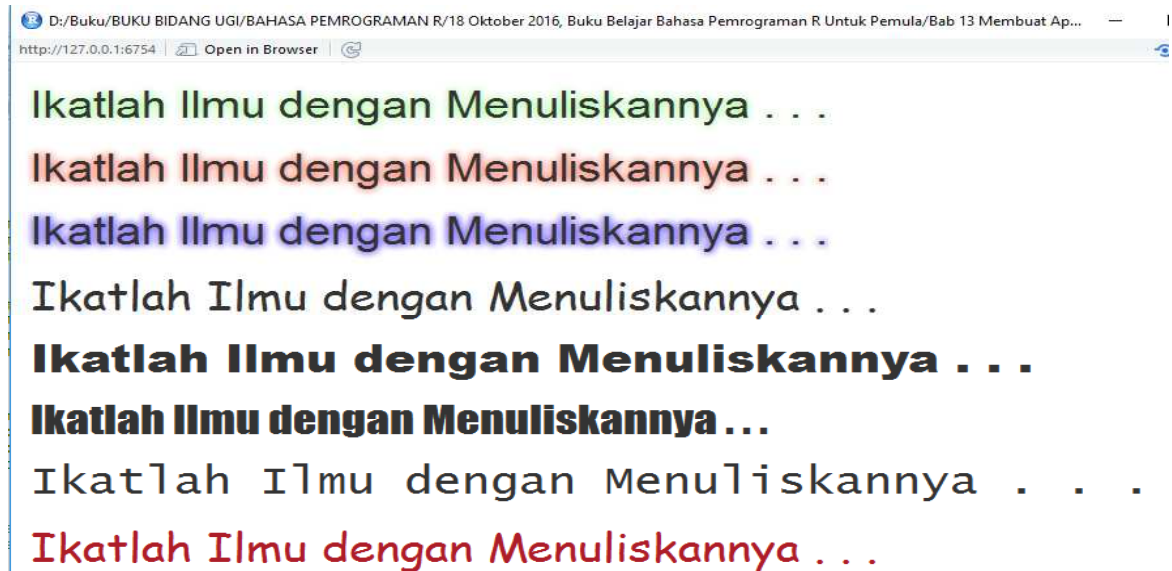
Prana Ugiana Gio

Prana Ugiana Gio

```
1 library(shiny)
2
3 shinyUI(
4
5   fluidPage(
6
7     h1("Prana Ugiana Gio", style="text-shadow:2px 2px; color:red;" ),
8     h1("Prana Ugiana Gio", style="text-shadow:5px 2px; color:green;" ),
9     h1("Prana Ugiana Gio", style="text-shadow:2px 5px; color:blue;" ),
10    h1("Prana Ugiana Gio", style="text-shadow:2px 2px 5px purple;" ),
11    h1("Prana Ugiana Gio", style="text-shadow:2px 20px 1px orange;font-style:italic;" ),
12    h1("Prana Ugiana Gio", style="text-shadow: -1px 0 black,
13      0 1px black, 1px 0 black, 0 -1px black; color:yellow"),
14
15    h1("Prana Ugiana Gio", style="box-shadow:5px 5px; width:700px; background-color:yellow;
16      text-shadow: -1px 0 black,
17      0 1px black, 1px 0 black, 0 -1px black; color:green"),
18
19    h1("Prana Ugiana Gio", style="box-shadow:5px 5px; width:700px; background-color:yellow;
20      text-shadow: -1px 0 black,
21      0 1px black, 1px 0 black, 0 -1px black; color:red;
22      font-size:80px;"),
23
24    h1("Prana Ugiana Gio", style="box-shadow:5px 15px blue; width:700px; background-color:yellow;
25      text-shadow: -1px 0 black,
26      0 1px black, 1px 0 black, 0 -1px black; color:red;
27      font-size:80px;"),
28
29
30    br()
31
32  ) #akhir fluidPage
33 ) #tutup kurung dari shinyUI
```

```
1 library(shiny)
2
3 shinyServer(
4
5   function(input, output) {
6
7   }
8 )
```

Gambar 13.45



```

1 library(shiny)
2
3 shinyUI(
4
5   fluidPage(
6
7     h1("Ikatlah Ilmu dengan Menuliskannya . . .", style="text-shadow: 0 0 0.2em #8F7"),
8     h1("Ikatlah Ilmu dengan Menuliskannya . . .", style="text-shadow: 0 0 0.2em #F87, 0 0 0.2em #F87"),
9     h1("Ikatlah Ilmu dengan Menuliskannya . . .", style="text-shadow: 0 0 0.2em #87F, 0 0 0.2em #87F,
10        0 0 0.2em #87F"),
11
12     h1("Ikatlah Ilmu dengan Menuliskannya . . .", style="font-family:Comic Sans MS"),
13     h1("Ikatlah Ilmu dengan Menuliskannya . . .", style="font-family:Arial Black"),
14     h1("Ikatlah Ilmu dengan Menuliskannya . . .", style="font-family:Impact"),
15     h1("Ikatlah Ilmu dengan Menuliskannya . . .", style="font-family:Lucida Console"),
16
17
18
19     h1("Ikatlah Ilmu dengan Menuliskannya . . .", style="font-family: 'Lobster', cursive;
20        font-weight: 500;
21        line-height: 1.1;
22        color: #ad1d28;"),
23
24     br()
25
26   ) #akhir fluidPage
27 ) #tutup kurung dari shinyUI
  
```

```

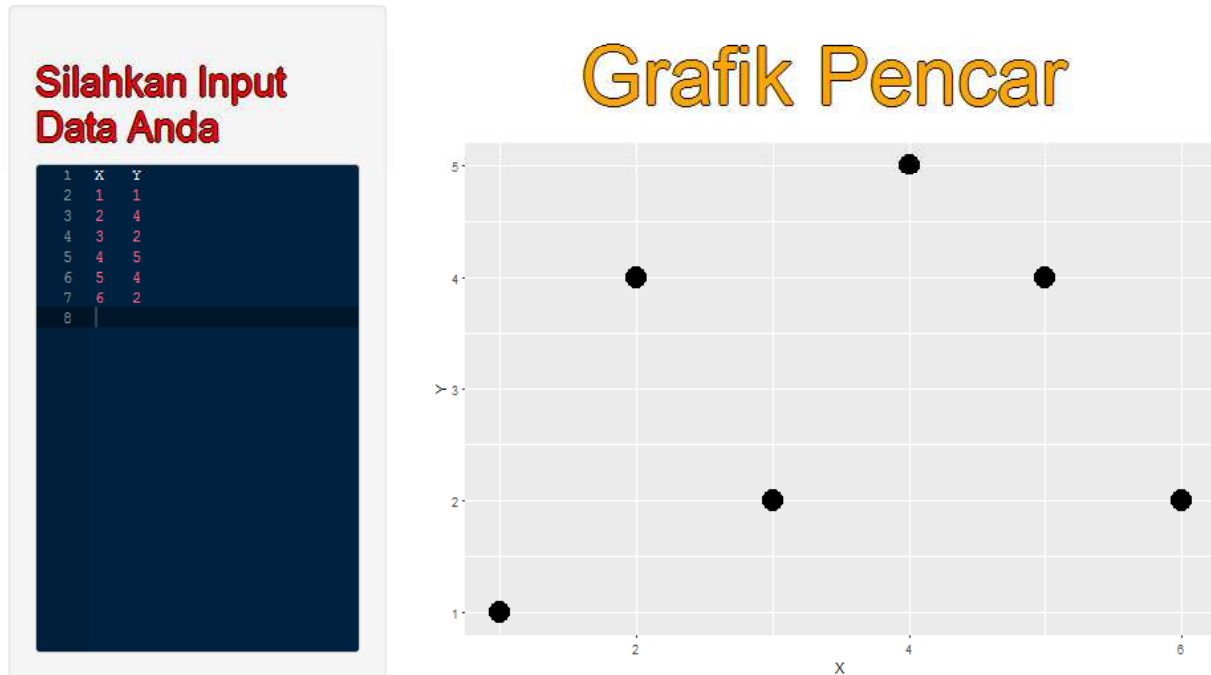
1 library(shiny)
2
3 shinyServer(
4
5   function(input, output) {
6
7   }
8 )
  
```

Gambar 13.46

13.6 Membuat Program Aplikasi Shiny (Praktek 6)

Sekarang akan dibuat program aplikasi shiny seperti pada Gambar 13.47. Program aplikasi shiny pada Gambar 13.37 dirancang untuk menyajikan data dari dua variabel (variabel **X** dan variabel **Y**) ke dalam grafik.

Program Aplikasi untuk Grafik Pencar



Gambar 13.47

Untuk membuat program aplikasi shiny tersebut, tulis kode R untuk **ui.r** dan **server.r** seperti pada Gambar 13.48 dan 13.49. Simpan **ui.r** dan **server.r** dalam satu folder. Kemudian jalankan aplikasi dengan meng-*click* **Run App**.

Berdasarkan Gambar 13.48, kode R

```
ambil_data <- read.csv(text=input$grafik_pencar, sep=" ", header = TRUE )
```

bertujuan untuk menampilkan tampilan seperti pada Gambar 13.50. Pada Gambar 15.50, diminta untuk mengisi data pada variabel **X** dan **Y**. Setelah mengisi data pada variabel **X** dan **Y**, maka secara otomatis akan muncul grafik pencar. Sementara kode R

```
plotOutput("grafik_pencar")
```

bertujuan untuk menampilkan grafik pencar. Instruksi dari **grafik_pencar** dapat dilihat pada bagian **server.r**, yakni

```
grafik_pencar <- function(){
  ambil_data <- read.csv(text=input$grafik_pencar, sep="", header = TRUE )
  ggplot(data=ambil_data, aes(x=X, y=Y)) + geom_point(size=7)
}

output$grafik_pencar <- renderPlot({
  print(grafik_pencar())
})
```

Pastikan sudah diinstal *Package shinyAce!!!* Aktifkan (*load*) *package shinyAce*, karena akan digunakan fungsi *aceEditor*.

```
1 library(shiny)
2 library(shinyAce)
3 library(ggplot2)
4
5 shinyUI(
6   pagewithsidebar(
7     headerPanel(span("Program Aplikasi untuk Grafik Pencar", style="color:blue;
8                       font-family:Lucida Console;
9                       box-shadow:5px 5px; width:700px; background-color:pink;
10                      font-size:40px" ) ,
11     sidebarPanel(
12       h3("silahkan input Data Anda", style="color:red;text-shadow: -1px 0 black,
13         0 1px black, 1px 0 black, 0 -1px black;font-size:30px"),
14       aceEditor("grafik_pencar", value="X\tY", mode="r", theme="cobalt", height = 360),
15       br(),
16     ),
17     mainPanel(
18       h3("Grafik Pencar", style="color:orange;text-shadow: -1px 0 black,
19         0 1px black, 1px 0 black, 0 -1px black;font-size:60px; text-align:center"),
20       plotOutput("grafik_pencar")
21     )
22   )
23 )
24 )
25 )
26 )
27 )
28 )
29 )
30 )
31 )
32 )
33 )
34 )
35 )
```

22:11 (Top Level) ↕

Gambar 13.48 Kode ui.r

```

1 library(shiny)
2 library(shinyAce)
3 library(ggplot2)
4
5 shinyServer (
6
7   function(input, output) {
8
9
10    grafik_pencar <- function(){
11
12
13      ambil_data <- read.csv(text=input$grafik_pencar, sep="", header = TRUE )
14
15      ggplot(data=ambil_data, aes(x=X, y=Y)) + geom_point(size=7)
16
17    }
18
19    output$grafik_pencar <- renderPlot({
20      print(grafik_pencar())
21    })
22
23
24
25 }
26 )

```

Pastikan sudah diinstal *Package shinyAce!!!* Aktifkan (*load*) *package shinyAce*, karena akan digunakan fungsi *aceEditor*.

Gambar 13.49 Kode server.r



Gambar 13.50

Kode R

```
ambil_data <- read.csv(text=input$grafik_pencar, sep="", header = TRUE )
```

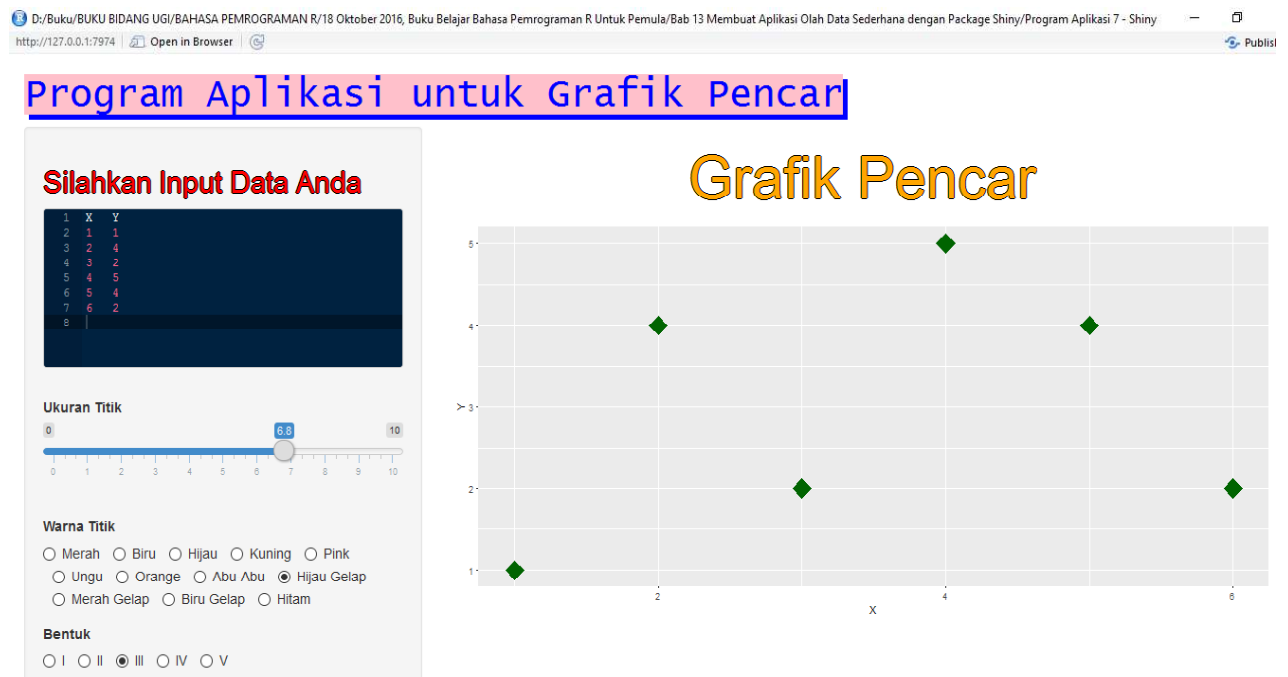
bertujuan mendapatkan data variabel X dan Y, kemudian ditugaskan disimpan di **ambil_data**. Selanjutnya kode R

```
ggplot(data=ambil_data, aes(x=X, y=Y)) + geom_point(size=7)
```

bertujuan untuk menampilkan grafik pencar, dengan ukuran titik sebesar 7 (**size = 7**).

13.7 Membuat Program Aplikasi Shiny (Praktek 7)

Sekarang akan dibuat program aplikasi shiny seperti pada Gambar 13.51. Program aplikasi shiny pada Gambar 13.51 merupakan pengembangan dari program aplikasi shiny pada Gambar 13.47.



Gambar 13.51

Untuk membuat program aplikasi shiny tersebut, tulis kode R untuk **ui.r** dan **server.r** seperti berikut.

```

server.R *  ui.r *
1 library(shiny)
2 library(shinyAce)
3 library(ggplot2)
4
5 shinyUI(
6
7   pagewithSidebar(
8
9     headerPanel(span("Program Aplikasi untuk Grafik Pencar", style="color:blue;
10                    font-family:Lucida Console;
11                    box-shadow:5px 5px; width:700px; background-color:pink;
12                    font-size:40px")),
13
14     sidebarPanel(
15
16       h3("Silahkan Input Data Anda", style="color:red;text-shadow: -1px 0 black,
17         0 1px black, 1px 0 black, 0 -1px black;font-size:30px"),
18
19       aceEditor("grafik_pencar", value="X\tY", mode="r", theme="cobalt", height = 160),
20
21       br(),
22
23       sliderInput("ukuran_titik", label="ukuran Titik", min=0, max=10, value=3, step=0.2),
24
25       br(),
26
27       radioButtons("warna_titik", "warna Titik", c("Merah"="red", "Biru"="blue",
28         "Hijau"="green", "Kuning"="yellow", "Pink"="pink", "Ungu"="purple", "Orange"="orange",
29         "Abu-Abu"="grey", "Hijau Gelap"="darkgreen", "Merah Gelap"="darkred", "Biru Gelap"="darkblue",
30         "Hitam"="black"), inline = TRUE ),

```

Gambar 13.52 Kode ui.r

```

r.R x ui.r x
Run App
br(),
sliderInput("ukuran_titik", label="ukuran Titik", min=0, max=10, value=3, step=0.2),
br(),
radioButtons("warna_titik", "Warna Titik", c("Merah"="red", "Biru"="blue",
"Hijau"="green", "Kuning"="yellow", "Pink"="pink", "Ungu"="purple", "Orange"="orange",
"Abu-Abu"="grey", "Hijau Gelap"="darkgreen", "Merah Gelap"="darkred", "Biru Gelap"="darkblue",
"Hitam"="black"), inline = TRUE ),

radioButtons("bentuk", "Bentuk", c("I"="21",
"II"="22", "III"="23", "IV"="24", "V"="25"), inline = TRUE ),

br()
),
mainPanel(
h3("Grafik Pencar", style="color:orange;text-shadow: -1px 0 black,
0 1px black, 1px 0 black, 0 -1px black;font-size:60px; text-align:center"),
plotOutput("grafik_pencar")
)
)
)
(Top Level) R Script

```

Gambar 13.53 Sambungan Kode ui.r

```

server.R x ui.r x
1 library(shiny)
2 library(shinyAce)
3 library(ggplot2)
4
5 shinyServer(
6
7   function(input, output) {
8
9
10    grafik_pencar <- function(){
11
12
13      ambil_data <- read.csv(text=input$grafik_pencar, sep="", header = TRUE )
14
15      ukuran_titik <- input$ukuran_titik
16      warna_titik <- input$warna_titik
17      bentuk <- input$bentuk
18      bentuk = as.numeric(bentuk)
19
20      ggplot(data=ambil_data, aes(x=X, y=Y)) + geom_point(size=ukuran_titik,
21      colour=c(warna_titik), shape=c(bentuk), fill=c(warna_titik) )
22
23    }
24
25    output$grafik_pencar <- renderPlot({
26      print(grafik_pencar())
27    })
28
29
30
21:65 grafik_pencar()

```

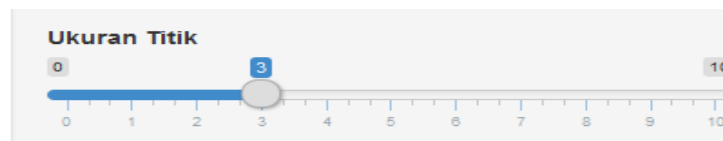
Gambar 13.54 Kode server.r

Simpan **ui.r** dan **server.r** dalam satu folder. Kemudian jalankan aplikasi dengan meng-*click* **Run App**.

Berdasarkan Gambar 13.52, kode R

```
sliderInput("ukuran_titik", label="Ukuran Titik", min=0, max=10, value=3, step=0.2),
```

bertujuan untuk menampilkan tampilan seperti pada Gambar 13.55, yang bertujuan untuk mengatur besar kecil ukuran titik.

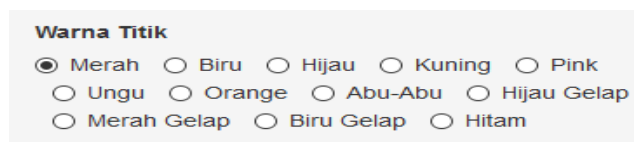


Gambar 13.55

Kode R

```
radioButtons("warna_titik", "Warna Titik", c("Merah"="red", "Biru"="blue",  
"Hijau"="green", "Kuning"="yellow", "Pink"="pink",  
"Ungu"="purple", "Orange"="orange",  
"Abu-Abu"="grey", "Hijau Gelap"="darkgreen", "Merah Gelap"="darkred", "Biru  
Gelap"="darkblue",  
"Hitam"="black"), inline = TRUE ),
```

bertujuan untuk menampilkan tampilan seperti pada Gambar 13.56, yang bertujuan untuk mengatur warna titik.

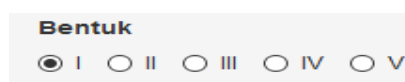


Gambar 13.56

Kode R

```
radioButtons("bentuk", "Bentuk", c("I"="21",  
"II"="22", "III"="23", "IV"="24", "V"="25"), inline = TRUE ),
```

Bertujuan untuk menampilkan tampilan seperti pada Gambar 13.57, yang bertujuan untuk mengatur bentuk (titik, segiempat, segitiga, dan sebagainya).



Gambar 13.57

13.8 Dasar-dasar Shiny

Shiny merupakan aplikasi berupa halaman web(UI) yang terhubung dengan sebuah komputer yang menjalankan aplikasi R(Server). Pengguna dapat memanipulasi, mengubah tampilan UI yang akan mengakibatkan server mengupdate tampilan UI dengan menjalankan sintak R.



Gambar 13.58

13.9 Template

Untuk mempelajari shiny, langkah pertama yang dapat dilakukan adalah memulai menulis dengan template yang sudah ada. Jalankan sintak berikut di command line R.

```
library(shiny)
ui <- fluidpage()
server <- function(input,output){}
shinyApp(ui = ui, server = server)
```

Berikut penjelasan dari sintaks di atas,

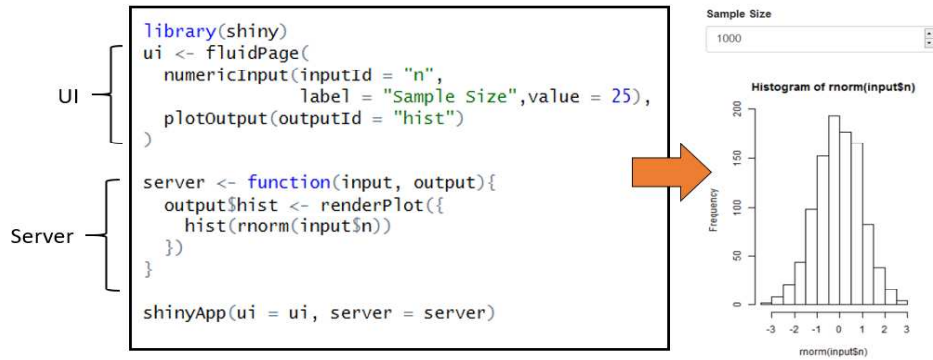
- ui** - fungsi R yang menghimpun komponen-komponen HTML untuk tampilan depan aplikasi
- server** - sebuah fungsi dengan instruksi-instruksi yang akan merencanakan objek-objek yang ditampilkan pada UI.
- shinyApp** - mengombinasikan **ui** dan **server** menjadi sebuah aplikasi.

13.10 Membuat Sebuah Aplikasi

Selanjutnya, akan diuraikan bagaimana menyempurnakan sintak di sub-bab 2.2. Pada bagian UI, telah dilakukan penambahan fungsi masukan ***Input()**, dan penambahan keluaran dengan fungsi ***Output()**. Untuk memberi komando kepada server supaya merubah (*render*) tampilan output, maka diberi perintah di bagian Server,

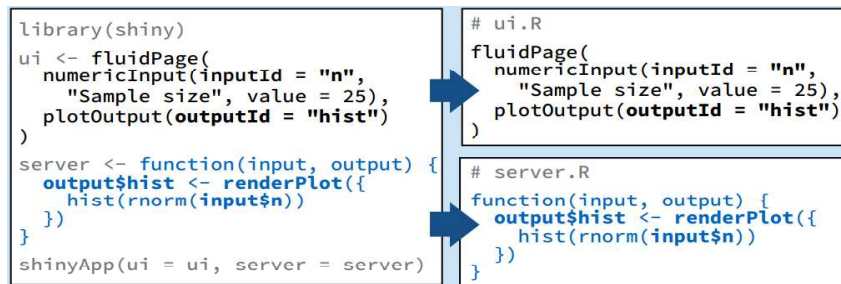
1. Mengacu ke *outputs* dengan `output$<id>`
2. Mengacu ke *inputs* dengan `input$<id>`

Sebelum outputnya di save, sintaks2 yang merupakan *output* di *wrap* dalam sintak **render*()**



Gambar 13.59

Simpan template anda sebagai **app.R**. Selain itu, kita juga dapat memisahkan template menjadi dua file terpisah. Namakan **ui.R** dan **server.R**. **ui.R** berisi segala sesuatu yang dibutuhkan untuk membangun tampilan, sementara **server.R** berisi fungsi-fungsi yang dibutuhkan untuk mengubah tampilan ketika diberi perubahan inputan.



Gambar 13.60

13.11 Inputs

Mengumpulkan nilai-nilai dari pengguna(*user*) dengan **input\$<inputId>**. Nilai-nilai dari input merupakan *reactive*, digambarkan berikut:

Buttons <input type="button" value="Action"/> <input type="button" value="Submit"/> actionButton() submitButton()	Single checkbox <input checked="" type="checkbox"/> Choice A checkboxInput()	Checkbox group <input checked="" type="checkbox"/> Choice 1 <input type="checkbox"/> Choice 2 <input type="checkbox"/> Choice 3 checkboxGroupInput()	Date input <input type="text" value="2014-01-01"/> dateInput()
Date range <input type="text" value="2014-01-24"/> to <input type="text" value="2014-01-24"/> dateRangeInput()	File input <input type="button" value="Choose File"/> No file chosen fileInput()	Numeric input <input type="text" value="1"/> numericInput()	Password Input <input type="password" value="....."/> passwordInput()
Radio buttons <input checked="" type="radio"/> Choice 1 <input type="radio"/> Choice 2 <input type="radio"/> Choice 3 radioButtons()	Select box <input type="text" value="Choice 1"/> selectInput()	Sliders <input type="range" value="50"/> sliderInput()	Text input <input type="text" value="Enter text..."/> textInput()

Gambar 13.61

13.12 Outputs

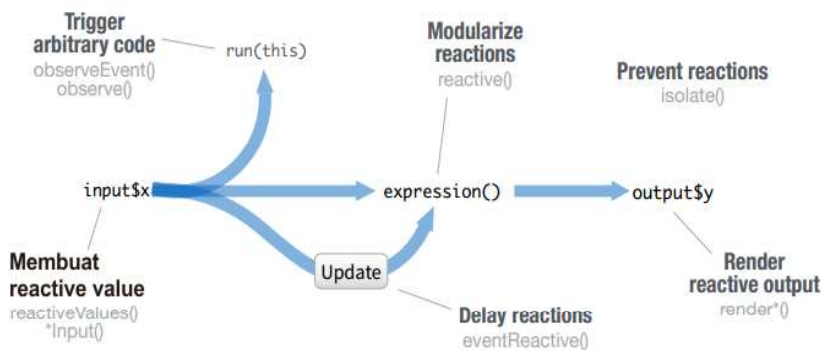
Fungsi `render*()` dan `*Output()` berjalan bersama untuk menampilkan output R ke UI.



Gambar 13.62

13.13 Reactivity

Nilai-nilai *reactive* berjalan bersama-sama dengan fungsi *reactive*. Mengakses nilai *reactive* dari argument fungsi-fungsi di bawah ini untuk menghindari bunyi error **Operation not allowed without an active reactive context.**



Gambar 13.63

13.13.1 Membuat Reactive Value

Pada shiny, reactive merupakan cara untuk membuat variabel pada bagian server. Fungsi `*Input()` menghasilkan reactive value yang tersimpan di `input$<inputId>`. Selain menggunakan `*input`, cara lain mendefinisikan variabel di server yakni menggunakan `reactiveValues()`, yang akan menghasilkan list variabel yang kita inginkan.

```

#Create Your Own Reactive Values
ui <- fluidPage(
  textInput("a",""),
  textOutput(outputId = "siang")
)

server <- function(input,output){
  rv <- reactiveValues()
  rv$number <- 5
  rv$saya <- 10

  output$siang <- renderText({
    paste(input$a,rv$saya, rv$number)
  })
}

shinyApp(ui = ui, server = server)

```

Gambar 13.64

13.13.2 Trigger Arbitrary Code

Kita dapat melihat hasil dari perintah yang dibuat supaya muncul di console R, seperti di bawah ini.

```

ui <- fluidPage(
  textInput("a",""),
  actionButton("go","Klik Ini")
)

server <- function(input,output){
  observeEvent(input$go,
    print(input$a))

  values <- reactiveValues(A = 1)
  obsB <- observe({
    print(values$A + 1)
  })

  obsC <- observe(quote({
    print(values$A + 2)
  }), quoted = TRUE)
}

shinyApp(ui, server)

```

Gambar 13.65

13.13.3 Delay Reaction

Untuk mengasilkan suatu program yang memiliki tampilan(*interface*) yang interaktif, kita dapat menggunakan fitur *delay reaction*. Fitur ini akan menjalankan code setelah menunggu *action* dari user.

13.13.4 Modularize Reactions

Bedanya **reactiveValues()** dengan **reactive()** ialah hasil tipe data yang dihasilkan. Jika **reactiveValues()** akan menghasilkan list, sementara **reactive()** dapat digunakan untuk membuat perintah-perintah yang akan digunakan, sehingga tidak perlu diulang pada setiap badan code Output.

```

ui <- fluidPage(
  textInput("a",""),
  textInput("z",""),
  textOutput("b")
)

server <- function(input, output){
  re <- reactive({
    paste(input$a, input$z)
  })

  output$b <- renderText({
    re()
  })
}

shinyApp(ui = ui, server = server)

```

Gambar 13.66

13.13.5 Render Reactive Output

Fitur ini akan menghasilkan object yang akan ditampilkan. Sesuai namanya yakni *render*, akan menjalankan ulang code untuk menghasilkan ulang object kapanpun nilai reactive berubah.

```

ui <- fluidPage(
  textInput(inputId = "a",label = "Masukkan Teks"),
  textOutput("b")
)

server <- function(input, output){
  output$b <- renderText({
    input$a
  })
}

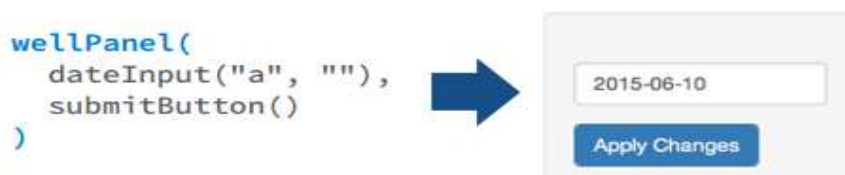
shinyApp(ui = ui, server = server)

```

Gambar 13.67

13.14 Layout

Ada banyak tampilan yang dapat digunakan pada shiny. Jika membicarakan tampilan, maka fokus kita tertuju pada code di bagian UI. Inputan-inputan yang digunakan dinamakan element. Kita dapat mengkombinasikan beberapa element menjadi satu elemen dengan fungsi Panel, seperti contoh berikut,



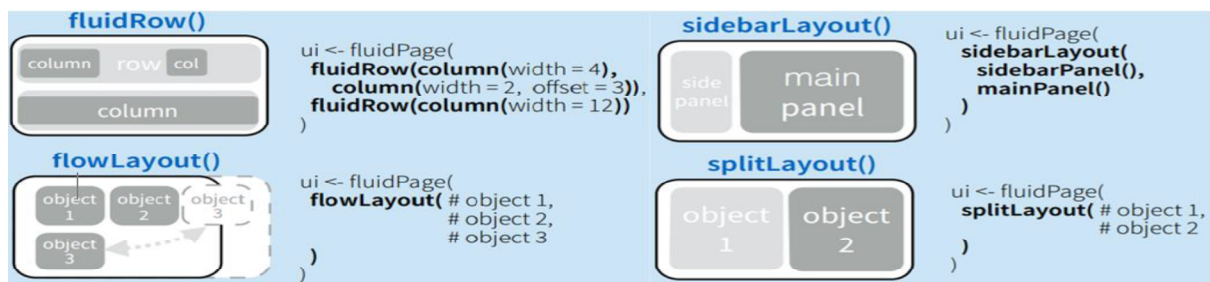
Gambar 13.68

Selain wellpanel, ada banyak yang penulis sarankan pembaca untuk mencoba sendiri hal-hal berikut ini. Cukup ketikkan di help Rstudio.



Gambar 13.69

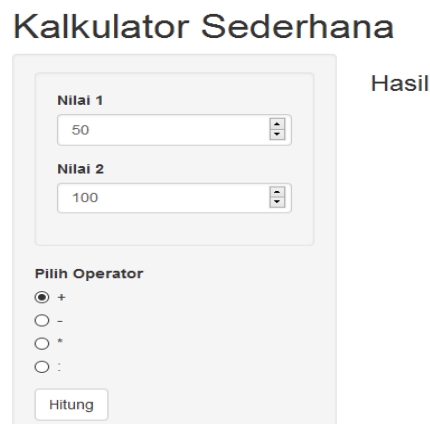
Kita juga dapat mengatur panel dan element dengan menggunakan fungsi layout. Dengan menjadikan elemen-elemen sebagai argumen dari fungsi layout.



Gambar 13.70

13.15 Membuat Kalkulator Sederhana

Pada bagian ini akan dibahas langkah-langkah membuat kalkulator yang sederhana seperti gambar di bawah ini,



Gambar 13.71

13.15.1 Mendesain UI

Untuk membuat program kalkulator seperti gambar di atas, elemen-elemen yang dibutuhkan ialah,

- Model Layout dengan *sidebarPanel* dan *MainPanel*
- Elemen *input* yakni *numericInput*

- Kedua *input*-an di jadikan satu dengan menggunakan *wellPanel*
- Untuk pilihan operator perhitungan, digunakan element *radioButton*
- Supaya ada interaksi antara pengguna dan program, maka digunakan element *actionButton*

13.15.2 Menyusun Sintaks

Sintaks pada bagian server disajikan di bawah berikut ini,

```

ui <- fluidPage(
  h1("Kalkulator Sederhana"),
  sidebarLayout(
    sidebarPanel(
      wellPanel(
        numericInput("masukan1", "Nilai 1", value = 50),
        numericInput("masukan2", "Nilai 2", value = 100),
        radioButtons(inputId = "operator", label = "Pilih Operator",
                    choices = c("+", "-", "*", ":") ),
        actionButton("a", "Hitung")),
      mainPanel(
        h3("Hasil"),
        textOutput("ab")
      )
    )
  )
server <- function(input, output){
  ide <- eventReactive(
    input$a, {
      A = input$masukan1
      B = input$masukan2
      if(input$operator == "+"){
        A + B
      }else if(input$operator == "-"){
        A - B
      }else if(input$operator == "*"){
        A*B
      } else{ A / B}
    }
  )
  output$ab <- renderText({
    ide()
  })
}
shinyApp(ui, server)

```

Gambar 13.72

Referensi Bacaan

- ⇒ Andrie de Vries dan Joris Meys, 2012, *R for Dummies*, John Willey & Sons
- ⇒ Garrett Golemund, 2014, *Hands-On Programming with R*, O'Reilly Media, Inc.,
- ⇒ Nina Zumel dan John Mount, 2014, *Practical Data Science with R*, Manning Publications
- ⇒ Norman Matloff, 2009, *The Art of R Programming*
- ⇒ Phil Spector, 2008, *Data Manipulation with R*, Springer
- ⇒ Prana Ugiana Gio dan Dasapta Erwin Irawan, 2016, Belajar Statistika dengan R
- ⇒ W. John Braum dan Duncan J. Murdoch, 2007, *A First Course in Statistical Programming with R*, Cambridge University Press
- ⇒ <https://shiny.rstudio.com/>
- ⇒ <http://deanattali.com/>
- ⇒ <http://langtest.jp/shiny/npt/>